FANUC Robot series

R-30*i*B/R-30*i*B Mate/R-30*i*B Plus/R-30*i*B Mate Plus/
R-30*i*B Compact Plus/R-30*i*B Mini Plus CONTROLLER

OPERATOR'S MANUAL (Basic Function)

B-83284EN/09

• Original Instructions

Thank you very much for purchasing FANUC Robot.

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

The products in this manual are controlled based on Japan’s “Foreign Exchange and Foreign Trade Law”. The export from Japan may be subject to an export license by the government of Japan.
Further, re-export to another country may be subject to the license of the government of the country from where the product is re-exported. Furthermore, the product may also be controlled by re-export regulations of the United States government.
Should you wish to export or re-export these products, please contact FANUC for advice.

In this manual, we endeavor to include all pertinent matters. There are, however, a very large number of operations that must not or cannot be performed, and if the manual contained them all, it would be enormous in volume. It is, therefore, requested to assume that any operations that are not explicitly described as being possible are "not possible".

SAFETY PRECAUTIONS

This chapter must be read before using the robot.

For detailed functions of the robot operation, read the relevant operator's manual to understand fully its specification.

For the safety of the operator and the system, follow all safety precautions when operating a robot and its peripheral equipment installed in a work cell.
For safe use of FANUC robots, you must read and follow the instructions in “FANUC Robot SAFETY HANDBOOK (**B-80687EN)**”.

1 DEFINITION OF USER

The personnel can be classified as follows.

Operator:
- Turns the robot controller power on/off
- Starts the robot program from operator panel

Programmer or Teaching operator:
- Operates the robot
- Teaches the robot inside the safety fence

Maintenance technician:
- Operates the robot
- Teaches the robot inside the safety fence
- Performs maintenance (repair, adjustment, replacement)

- Operator is not allowed to work in the safety fence.
- Programmer/Teaching operator and maintenance technician is allowed to work in the safety fence. Works carried out in the safety fence include transportation, installation, teaching, adjustment, and maintenance.
- To work inside the safety fence, the person must be trained on proper robot operation.

Table 1 (a) lists the work outside the safety fence. In this table, the symbol “○” means the work allowed to be carried out by the worker.

Table 1 (a) List of work outside the fence

	Operator	Programmer or Teaching operator	Maintenance technician
Turn power ON/OFF to Robot controller	○	○	○
Select operating mode (AUTO, T1, T2)		○	○
Select remote/local mode		○	○
Select robot program with teach pendant		○	○
Select robot program with external device		○	○
Start robot program with operator’s panel	○	○	○
Start robot program with teach pendant		○	○
Reset alarm with operator’s panel		○	○
Reset alarm with teach pendant		○	○
Set data on teach pendant		○	○
Teaching with teach pendant		○	○
Emergency stop with operator’s panel	○	○	○
Emergency stop with teach pendant	○	○	○
Operator’s panel maintenance			○
Teach pendant maintenance			○

In the robot operating, programming and maintenance, the operator, programmer/teaching operator and maintenance technician take care of their safety using at least the following safety protectors.

- Use clothes, uniform, overall adequate for the work
- Safety shoes
- Helmet

2 DEFINITION OF SAFETY NOTATIONS

To ensure the safety of users and prevent damage to the machine, this manual indicates each precaution on safety with "**WARNING**" or "**CAUTION**" according to its severity. Supplementary information is indicated by "**NOTE**". Read the contents of each "**WARNING**", "**CAUTION**" and "**NOTE**" before using the robot.

Symbol	Definitions
⚠WARNING	Used if hazard resulting in the death or serious injury of the user will be expected to occur if he or she fails to follow the approved procedure.
⚠CAUTION	Used if a hazard resulting in the minor or moderate injury of the user, or equipment damage may be expected to occur if he or she fails to follow the approved procedure.
NOTE	Used if a supplementary explanation not related to any of WARNING and CAUTION is to be indicated.

- Check this manual thoroughly, and keep it handy for the future reference.

TABLE OF CONTENTS

1 PREFACE

This chapter explains the manual plan at the usage of FANUC robot.

Contents of this chapter

1.1 ABOUT MANUAL

About this manual

"FANUC Robot series OPERATOR'S MANUAL(Basic Operation) B-83284EN" describes how to operate the FANUC Robot. It is controlled by the FANUC R-30*i*B, R-30*i*B Mate, R-30*i*B Plus, R-30*i*B Mate Plus, R-30*i*B Compact Plus and R-30*i*B Mini Plus controllers (called the robot controller hereinafter) containing the Application tool software.

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

- Setting the robot system
- Operating the robot
- Creating and changing a program
- Executing a program
- Robot status indications
- Backup and restore robot programs.

Using this manual

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

Chapter	Descriptions
Chapter 1 PREFACE	Describes how to use this manual.
Chapter 2 OVERVIEW	Gives a basic knowledge of the robot. It describes the basic configuration of the robot and the system for Robot.
Chapter 3 SETTING UP THE ROBOT SYSTEM	Describes the procedure for setting the system for Robot including input/output, coordinate system, and reference position.
Chapter 4 PROGRAM STRUCTURE	Describes the program structure and the syntax of program instructions.
Chapter 5 PROGRAMMING	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 DISPLAY	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, and so on.
Chapter 10 PALLETIZING FUNCTION	Describes the setting and executing of palletizing function.

Chapter	Descriptions
Chapter 11 APPLICATION FUNCTION OF TEACH PENDANT	Describes the setting and operation for the application function of the teach pendant.
Chapter 12 *i*HMI	Describes the function of *i*HMI details and operation method
Chapter 13 OPERATIONS WITHOUT A TEACH PEANDANT	Describes operations for controller without a teach pendant.
Chapter 14 OPERATION WITHOUT MODE SWITCH	Describes operations for controller without a mode switch.
APPENDIX	Describes lists of the menus, screens, program instructions and detail of program, special operations, lists of the system variables, saving research data.

Identification

For software version, read the following sections:

Item to be checked	Section
Verification of your software version	B.3 SOFTWARE VERSION

Specifications of products

For memory statuses, see the following sections:

Item to be checked	Section
Verification of memory status	7.12 MEMORY USE STATUS DISPLAY

Related manuals

The following FANUC Robot series manuals are available:

Robot controller	OPERATOR’S MANUAL (Basic Operation) **B-83284EN** (This manual)	Topics: Functions, operations and the procedure for operating the robot. Programming procedure and interface. Use: Guide to teaching, introducing, and adjusting the robot at the work site, and application designing.
	OPERATOR'S MANUAL (Alarm code list) **B-83284EN-1**	Topics: Error code listings, causes, and remedies. Use: Installing and activating the system, connecting the mechanical unit to the peripheral device and maintenance the robot.
	Optional Function OPERATOR’S MANUAL **B-83284EN-2**	Topics: Description of the software optional functions. Use: Guide to teaching, introducing, and adjusting the robot at the work site, and application designing.
	Arc Welding Function OPERATOR’S MANUAL **B-83284EN-3**	Topics: Description of the setting and operation for arc welding application software. Use: Guide to teaching, introducing, and adjusting the robot at the work site, and application designing.

Robot controller	Spot Welding Function OPERATOR'S MANUAL **B-83284EN-4**	Topics: Description of the setting and operation for spot welding application software. Use: Guide to teaching, introducing, and adjusting the robot at the work site, and application designing.
	Dispense Function OPERATOR'S MANUAL **B-83284EN-5**	Topics: Description of the setting and operation for dispense application software. Use: Guide to teaching, introducing, and adjusting the robot at the work site, and application designing.
	MAINTENANCE MANUAL **B-83195EN (for R-30*i*B, R-30*i*B Plus),** **B-83525EN (for R-30*i*B Mate, R-30*i*B Mate Plus)** **B-83555EN (for R-30*i*B Mate Open Air)** **B-84035EN (for R-30*i*B Compact Plus)** **B-84175EN (for R-30*i*B Mini Plus)**	Topics: Installing and activating the system, connecting the mechanical unit to the peripheral device and maintenance the robot.
Mechanical unit	OPERATOR'S MANUAL	Topics: Installing and activating the robot, connecting the mechanical unit to the controller, maintaining the robot. Use: Guide to installation, activation, connection, and maintenance.

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 Application tool software, the mechanical unit of the robot itself (FANUC Robot series), and the Robot controller.
The FANUC robot offers outstanding performance when handling or welding.

Tool software for application

The tool software for application is a software package for all kinds of Robot’s manipulations installed on the Robot controller. Any work can be performed by specifying menus and instructions from the teach pendant. The Application tool software contains instructions for controlling the robot, hands, remote controllers, and other peripheral devices.
The I/O between an additional axis or controller and another peripheral device can be controlled. Other peripheral devices include cell controllers or sensors.

Robot

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

Controller

The Robot controller supplies power to drive the mechanical unit.
The Application tool software is installed on the Robot controller to control the teach pendant, operator’s panel, and external peripheral devices.
Peripheral devices, including remote controllers, are required to configure a robot system for manipulating workpieces.

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

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

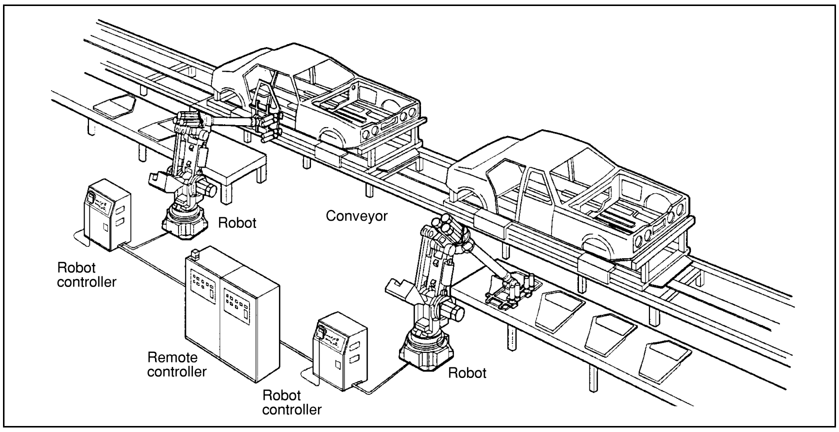


Fig. 2 Assembly system for car doors

2.1 APPLICATION TOOL SOFTWARE

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

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

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

2.1.1 System Setting

The Application tool software has an interface for various settings required to configure a robot system. (For setting up the Robot system, see Chapter 3.)
With the Application tool software, the hands, the 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. In order to teach a motion instruction in a program, the robot is moved to the target position by jog feed, then the position is recorded. (For moving the Robot by jog feed, see Subsection 5.2.3.)

2.1.3 Program

A program contains motion instructions, I/O instructions, register instructions, and branch instructions. (For the program structure, see Chapter 4.) Each instruction is assigned a statement number. The target work is accomplished by sequentially executing the instructions.
The teach pendant is used to create or correct a program. (For programming, see Chapter 5.) The program contains the following instructions. Fig.2.1.3 shows a basic program.

- **Motion instruction**: Moves the robot 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.
- **I/O instruction**: Sends or receives a signal to or from a peripheral unit.
- **Branch instruction**: Changes the flow of a program.
- **Wait instruction**: Holds execution of the program until the specified conditions are satisfied.
- **Routine call instruction**: Calls and executes a subprogram.
- **Macro instruction**: Calls a specified program and executes it.
- **Palletizing instruction**: Palletizes workpieces.
- **Program end instruction**: Terminates execution of a program.
- **Comment instruction**: Adds a comment to a program.
- **Other instructions**

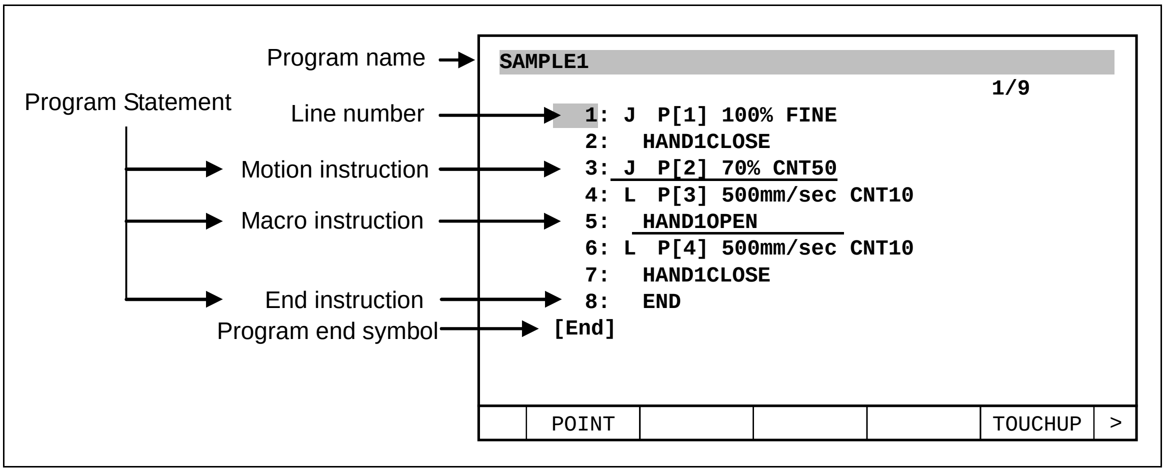


Fig. 2.1.3 Robot program

2.1.4 Test Operation (Test Execution)

After the system is set and a program is created, perform the test operation in the test execution mode to check the program for normal operation. (For the test operation, see Section 6.3.)
The test execution of the program is one of the important steps in creating a good program. Before starting automatic operation, perform the test execution of the created program.

2.1.5 Automatic Operation (Operation Execution)

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

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

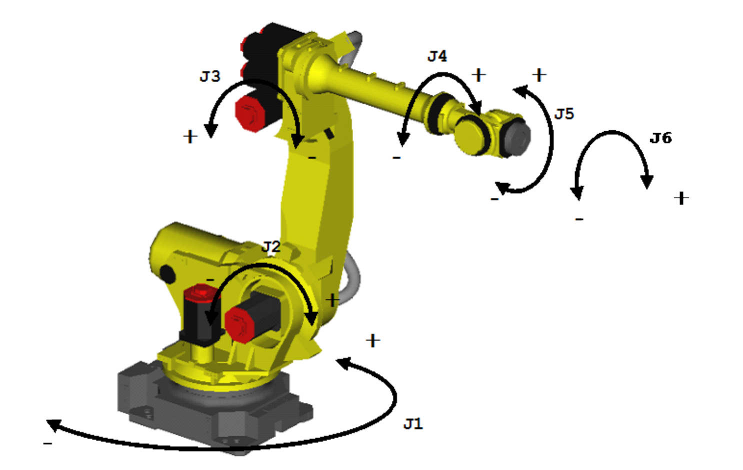


Fig. 2.2 (a) Main axes and wrist axes

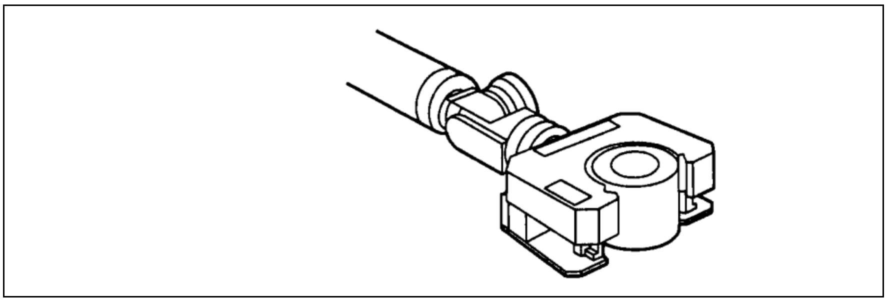


Fig. 2.2 (b) Hand with fingers

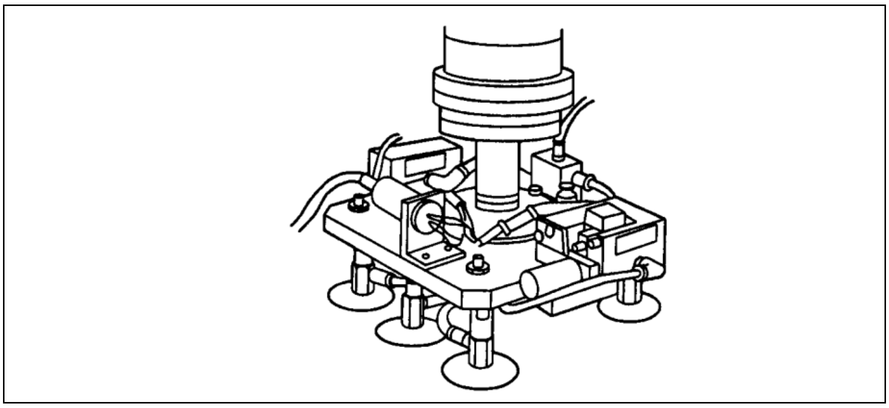


Fig. 2.2 (c) Hand with suction cups and no fingers

2.3 CONTROLLER

Robot controller includes a power unit, user interface circuit, motion controlling circuit, memory circuit, and input/output (I/O) circuit.
The user should use a teach pendant and operator's box to operate the controller.
The operation control circuit controls the servo amplifier which moves all the robot axes, including any additional axes, via the main CPU printed circuit board.
The memory circuit can store programs and data set by the user in the C-MOS RAM on the main CPU printed circuit board.
The I/O circuit interfaces the controller with the peripheral units by receiving and sending signals via the I/O modules (I/O printed circuit boards). The remote I/O signal is used for communication with the remote controller.

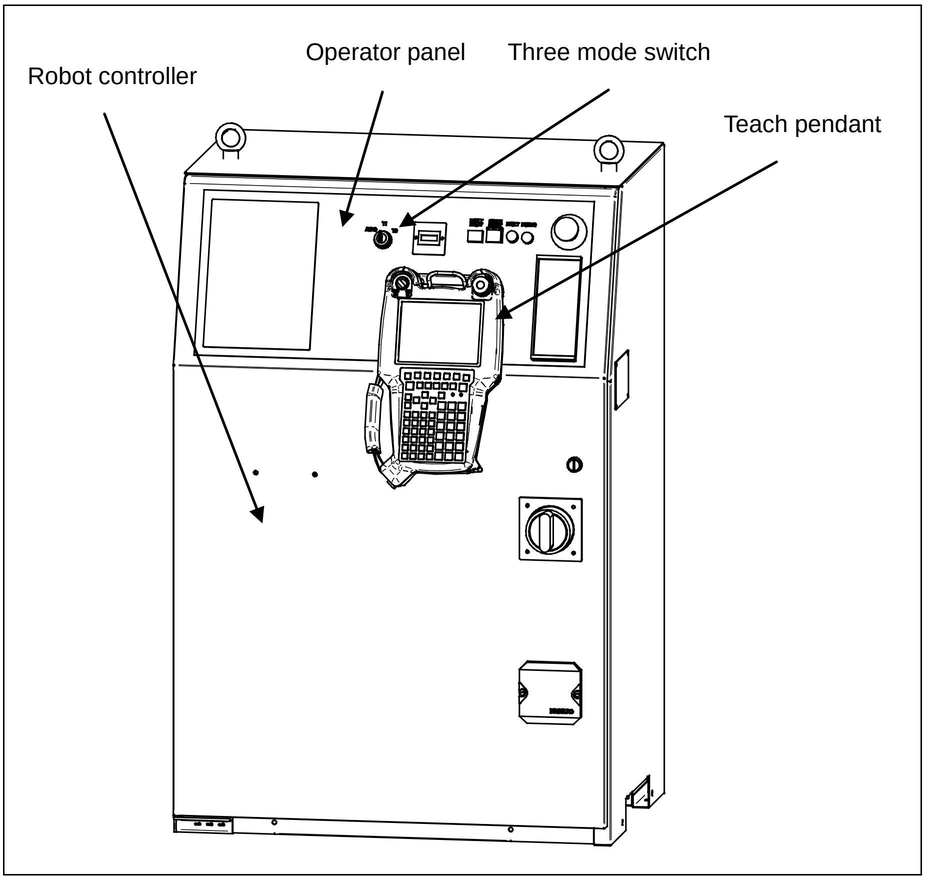


Fig. 2.3(a) Robot controller R-30*i*B

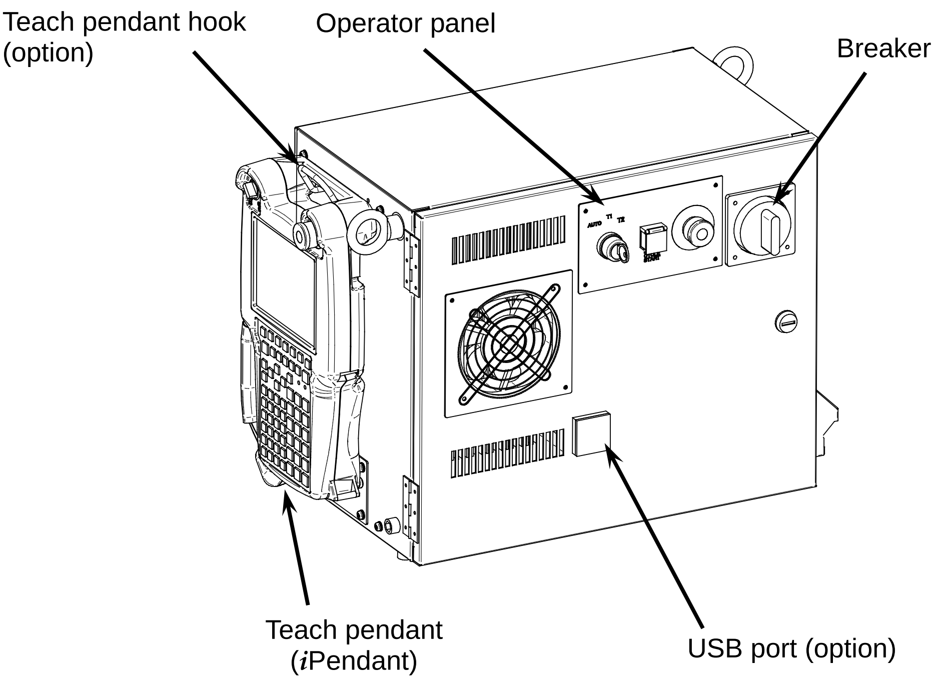


Fig. 2.3 (b) Robot controller R-30*i*B Mate

The circuitry of the controller depends on the robot and the system it controls. For details, refer to the “FANUC Robot series R-30*i*B/R-30*i*B Plus CONTROLLER MAINTENANCE MANUAL” (B-83195EN) or “FANUC Robot series R-30*i*B Mate/R-30*i*B Mate Plus CONTROLLER MAINTENANCE MANUAL” (B-83525EN) or "FANUC Robot series R-30*i*B Compact Plus CONTROLLER MAINTENANCE

MANUAL" (B-84035EN)" or "FANUC Robot series R-30*i*B Mini Plus CONTROLLER MAINTENANCE MANUAL" (B-84175EN)".

2.3.1 Teach Pendant

The teach pendant provides an interface between the Application tool software and the operator. The teach pendant is connected to the controller by a cable.
The teach pendant is optional for some controllers.

The following operations can be performed using the teach pendant:

- Jog feed of the robot
- Program creation
- Test execution of program
- Change settings
- Status check

The teach pendant includes the following:

- 640 x 480 pixel Liquid crystal display on R-30*i*B controller
- 1024 x 768 pixel Liquid crystal display on R-30*i*B Plus controller
- 2 LEDs
- 68 keys

NOTE

1. Design of the screen was changed in R-30*i*B Plus controller. In this manual the screen design is described at R-30*i*B controller.
2. On R-30*i*B Plus controller *i*HMI was added. For detail of *i*HMI, refer to "12 *i*HMI".

⚠ CAUTION

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

The following switches are also provided:

Table 2.3.1 (a) Switches on the teach pendant

Switch	Function
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	3 position DEATMAN SWITCH is enabled by pressing the switch to its midpoint. When the teach pendant is enabled, this switch allows robot motion only while the deadman switch is gripped. If you release this switch or grip this switch strongly, the robot stops immediately.
Emergency stop button	When pressed, the emergency stop button immediately stops the robot regardless of status of teach pendant enable switch (Please refer to "STOP TYPE OF ROBOT" in "SAFETY PRECAUTIONS" for detail of stop type).

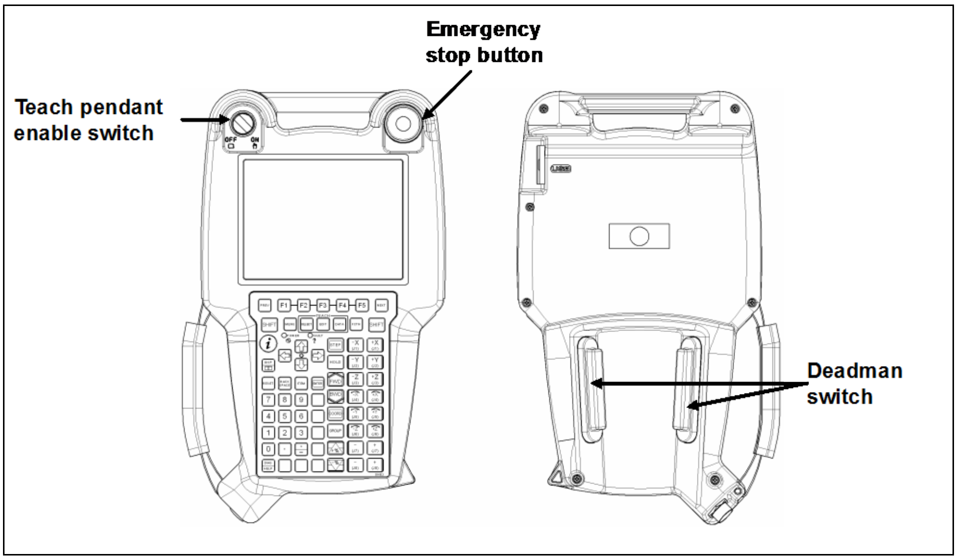


Fig. 2.3.1(a) Switches on the teach pendant

Fig. 2.3.1(b) and Fig. 2.3.1(c) show the examples of the key layout of the teach pendant.

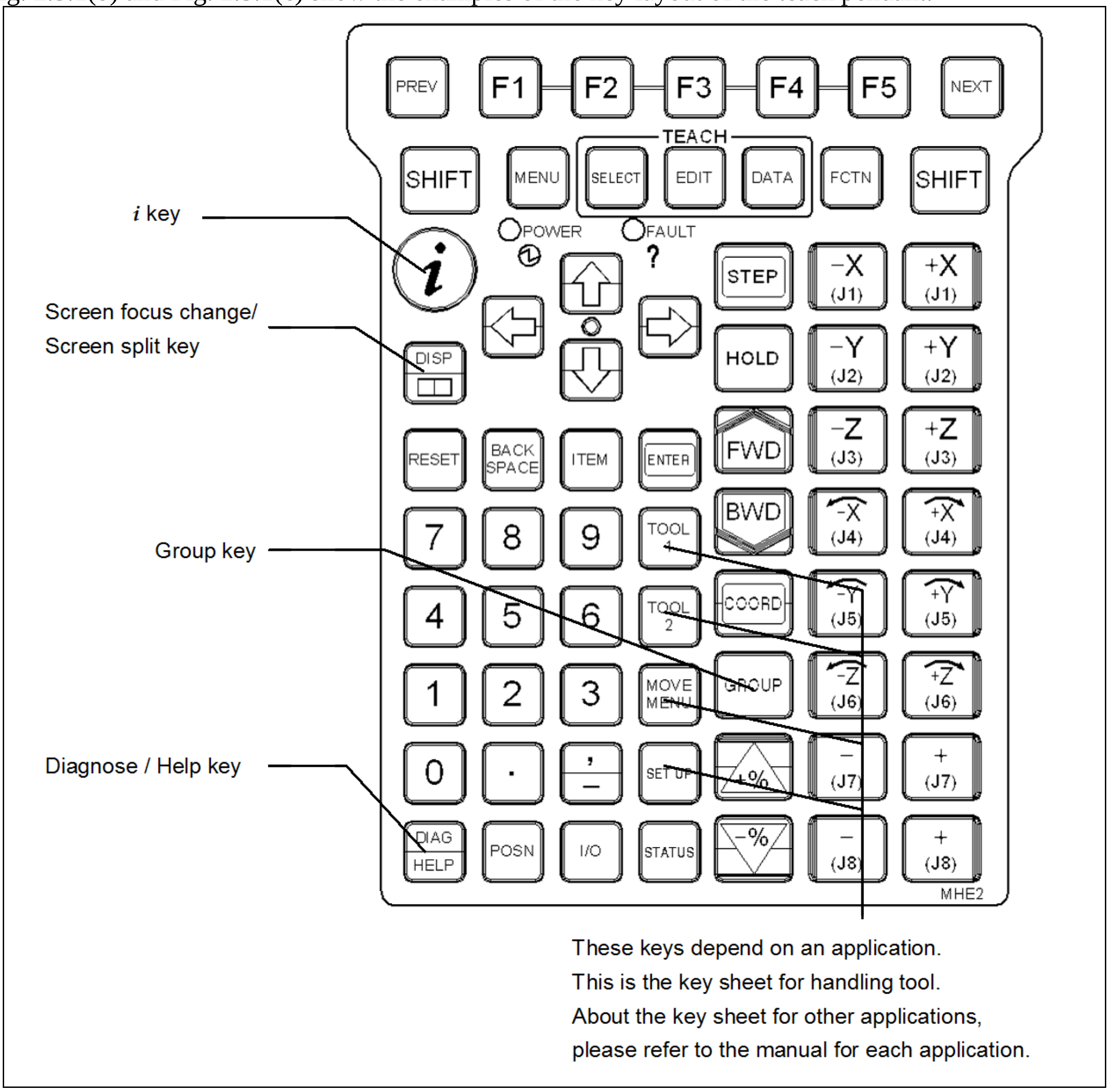


Fig. 2.3.1(b) Teach pendant keys (handling tool)

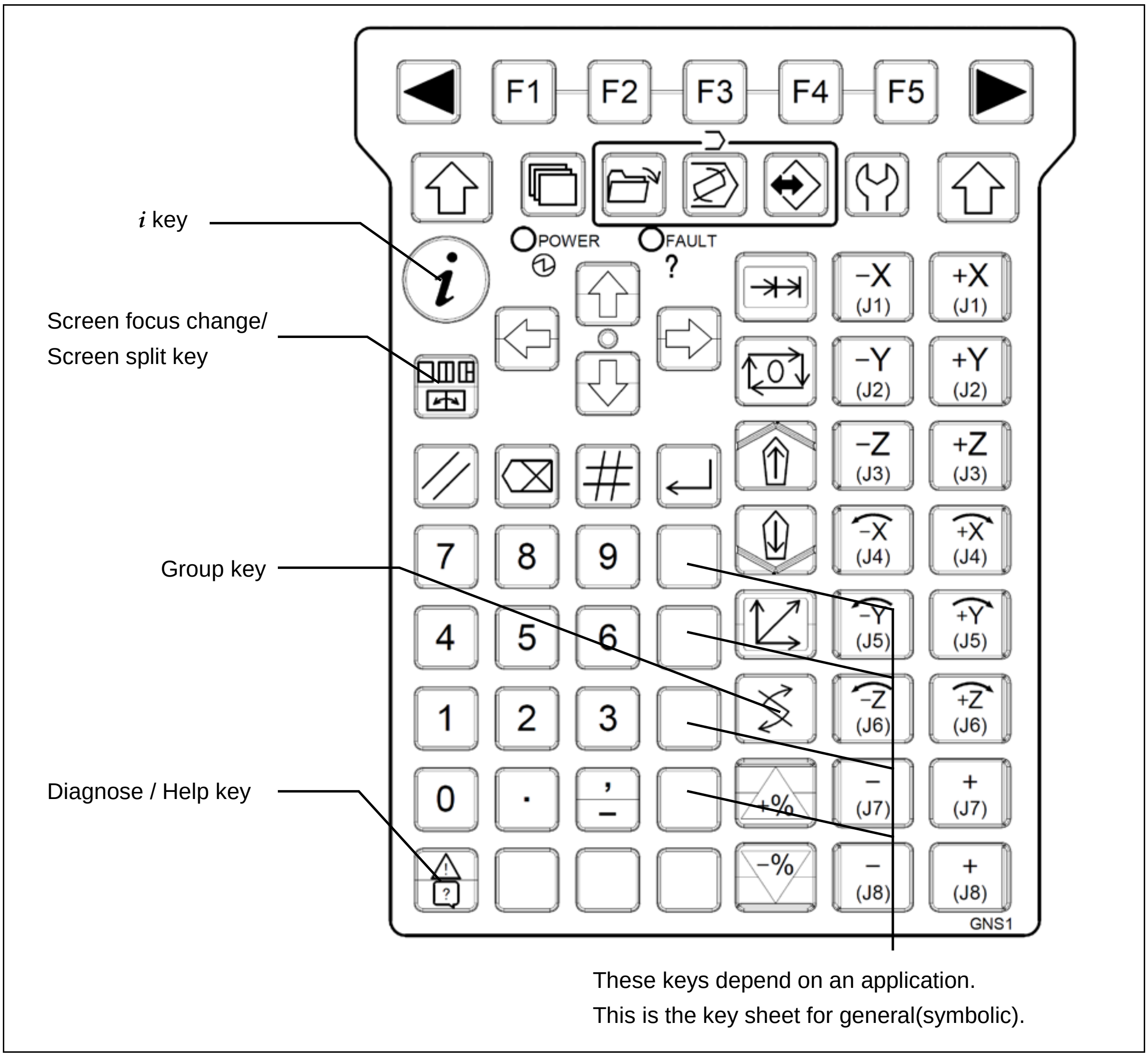


Fig. 2.3.1(c) Teach pendant keys (general(symbolic))

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
- Other keys

As Fig. 2.3.1(b) and Fig. 2.3.1(c), different key layouts have some keys with different words or pictures. But the key at the same position on each layout has the same function.

Table 2.3.1 (b) 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 or ▶	The next page key to switch the function key menu on the next page.

Key	Function
MENU or	The MENU key to display the screen menu.
FCTN or	The FCTN key to display the function menu.
SELECT or	The SELECT key to display the program selection screen.
EDIT or	The EDIT key to display the program edit screen.
DATA or	The DATA key to display the data screen.
TOOL 1 TOOL 2 or	These keys are application specific. In Handling Tool, the TOOL1 and TOOL2 key to display tool 1and 2 screen.
MOVE MENU or	This key is application specific. MOVE MENU key is not supported by Handling Tool.
SET UP or	This key is application specific. The SET UP key displays the setup screen.
STATUS or	The STATUS key displays the status screen.
I/O or	The I/O key displays the I/O screen.
POSN or	The POSN key displays the current position screen.
DISP or	In case that the screen on the teach pendant is split, when this key is pressed, the operation target screen is changed. When this key is pressed while [SHIFT] key is held down, the menu to split the screen is displayed.
DIAG HELP or	When this key is pressed, the hint screen is displayed. When this key is pressed while the [SHIFT] key is held down, the alarm screen is displayed.
GROUP or	Pressing this key changes the current motion group and sub-group selection step by step like this: G1, G1S, G2, G2S, G3, ... G1. Pressing a numeric key, that is identical with the group number you wish to select, with a GROUP key selects the wished motion group directly. You can toggle subgroup of currently selected group by pressing a 0 key with holding GROUP key pressed.

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

NOTE

The group key is enabled only when Multi motion group software option (J601) or Extended Axis Control software option (J518) has been ordered, and the extended axis or the independent axis has been set up.

Table 2.3.1 (c) Keys related to jog feed

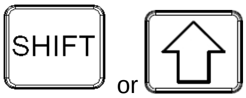
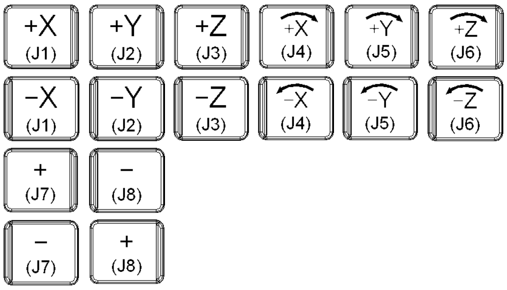
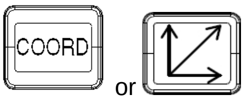
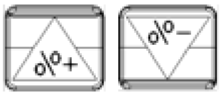
Key	Function
	The SHIFT key is used to execute a jog feed of the robot, teach the position data and start a program by pressing the key with another key. The right and left SHIFT keys have the same function.
	The jog keys are effective while [SHIFT] key is held down. They are used for jog feed. J7 key and J8 key are used for the jog feed of the extended axes in the same group. In case that the number of the robot axes is less than 6, the keys that is not used for the jog feed of the robot are used for the jog feed of the extended axes, too. Ex) In case that the number of the robot axes is 5, J6, J7 and J8 keys are used for the jog feed of the extended axes. The function of J7 and J8 key can be changed. Please refer to “Setting of J7, J8 key” in “5.2.3 Moving the Robot by Jog Feed”.
	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" and “JOINT”. When this key is pressed while [SHIFT] key is held down, a jog menu for changing the coordinate system appears.
	The override key adjusts the feed rate override. Each time the override key is pressed, it selects the next override in the order: "VFINE", "FINE", "1%, 5%, 50%, 100%". (changing amount 1% for 5% or less and changing amount 5% for 5% or more.)

Table 2.3.1 (d) Keys related to execution

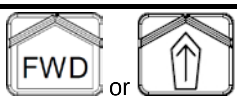
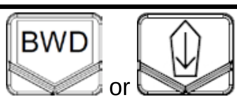
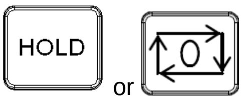
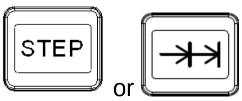
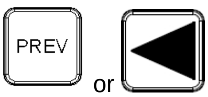
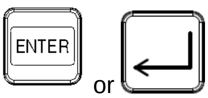
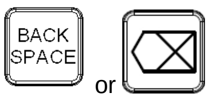
Key	Function
	The FWD key (+ [SHIFT] key) starts a program in normal order. When [SHIFT] key is released during the program execution, the program halts.
	The BWD key (+ [SHIFT] key) starts a program in reverse order. When [SHIFT] key is released during the program execution, the program halts.
	The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

Table 2.3.1 (e) Keys related to editing

Key	Function
	The PREV key restores the most recent state. In some cases, the screen may not return to the immediately preceding status.
	The ENTER key enters a numeral or selects a menu.
	The BACK SPACE key deletes the character or numeral immediately before the cursor.

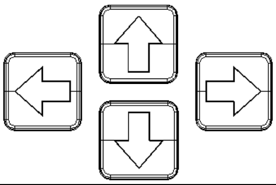
Key	Function
	The cursor key moves the cursor. The cursor is the highlighted part which can move on the teach pendant screen. This part becomes the object of operation (input or change of the value or contents) from the teach pendant key.
 or 	The ITEM key moves the cursor to a line whose number is specified.

Table 2.3.1 (f) Other keys

Key	Function
	*i* key can display Notifications screen on being pushed when blinking icon (Notification Icon) is displayed on status bar. Or it is used with the following keys. By pressing the following keys while the *i* key is held down, the operation by these keys is enhanced. For example, graphical screen is displayed. • MENU key • FCTN key • EDIT key • DATA key • POSN key • JOG key • DISP key

LEDs on the teach pendant

The teach pendant has the following two LEDs.

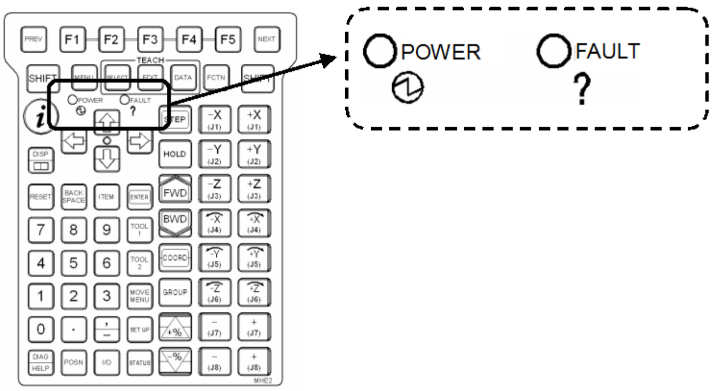


Fig. 2.3.1(d) LEDs on the teach pendant

Table 2.3.1 (g) LEDs on the teach pendant

LED	Description
POWER	The POWER LED indicates that the power of the controller is ON.
FAULT	The FAULT LED indicates that an alarm has occurred.

Touch panel

The teach pendant provides a touch panel as an option. The screens on which operations can be performed using the touch panel are as follows. Note that not all operations can be performed using the touch panel.

- Operation Panel screen / BROWSER screen (Web browser screen) / Status sub-window screen

- Software keyboard
- Screen switching (When multiple screens are displayed, moving to the desired screen is accomplished by touching the screen.), cursor movement
- Software button from F1 to F5 in the lower part of the screen

A beep occurs at the time of the use of the touch panel. To disable the beep, change the value of system variable $UI_CONFIG.$TOUCH_BEEP from TRUE to FALSE, and make restart (power off/on). This system variable can be used by system software 7DC2 (V8.20) series, version 01.or later.

NOTE

- If you press 2 or more places on touch panel, touch panel may recognize wrong place, which is different from touched place. Please touch only one place on touch panel.
- When touch panel is broken, the panel can recognize wrong place that are different from place operator touched. Don't use touch panel to set operations that can affect safety of system.
- Use touch panel by finger or dedicated pen for touch panel operation. If you use sharp object like usual pens, touch panel may get broken.
- When following dialog box is displayed on the teach pendant, touch panel may be broken. Turn down robot controller and exchange the teach pendants.

Status window

The window at the top of the teach pendant screen is called the status window. In this window, eight software LEDs, alarm indication, and override value are displayed.
Each software LED is "on" when displayed together with an icon or "off" when displayed with no icon.

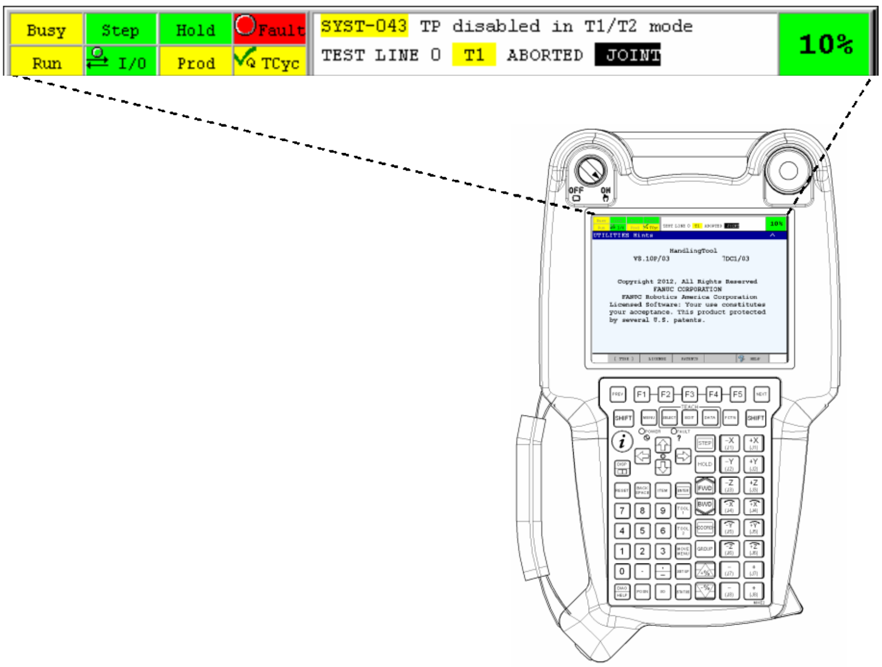


Fig. 2.3.1(e) Status window on the teach pendant

Table 2.3.1(h) Description of software LEDs

LEDs (Upper: On, Lower: Off)		Description
Busy	Busy Busy	Indicates that the robot is working.
Step	Step Step	Indicates that the robot is in the step operation mode.
Hold	Hold Hold	Indicates that the HOLD button is being held or the HOLD signal is input.
Fault	Fault Fault	Indicates that an alarm occurs.
Run	Run Run	Indicates that a program is being executed.
I/O	I/O I/O	Application-specific LED. This is a sample LED for a handling tool.
Prod	Prod Prod	Application-specific LED. This is a sample LED for a handling tool.
TCyc	TCyc TCyc	Application-specific LED. This is a sample LED for a handling tool.

Display screen of the teach pendant

The liquid crystal display screen (liquid crystal display) displays the Application tool software screen shown in Fig. 2.3.1(f). To operate the robot, select a screen corresponding to a desired function. The screen is selected by the screen menus shown in Fig. 2.3.1(g).

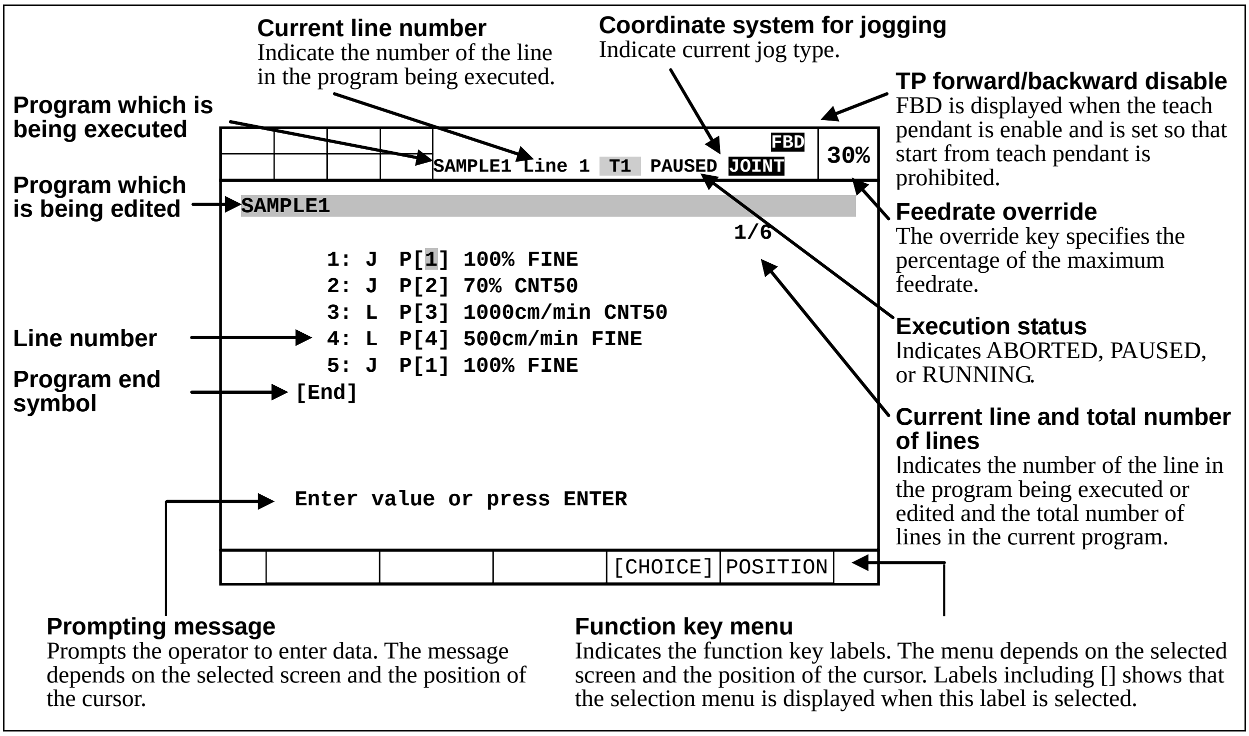


Fig. 2.3.1(f) Program edit screen

Menus

Menus are used to operate the teach pendant. The screen menu is displayed by [MENU] key and the function menu is displayed by [FCTN] key. And the top menu is displayed by the *i* key and [MENU] key. Fig. 2.3.1(g), Fig. 2.3.1(h), and Fig. 2.3.1(i) show the screen menu, the top menu, and the function menu respectively. And Fig. 2.3.1(j) shows the quick menu.

- 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.) To display the screen menu, press the [MENU] key on the teach pendant.

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

Page 1

```
        MENU 2
1 SELECT
2 EDIT
3 DATA
4 STATUS
5 4D GRAPHICS
6 SYSTEM
7 USER2
8 BROWSER
9
0 -- NEXT --
```

Page2

Fig. 2.3.1(g) Screen menu

Table 2.3.1(i) Screen menu items

Item	Description
UTILITIES	The utility screen is used to operate various functions.
TEST CYCLE	The test cycle screen is used to specify the data for test operation.
MANUAL FCTNS	Macro instructions are executed manually.
ALARM	Alarms currently posted, alarm history and details of them are displayed.
I/O	The I/O screen is used to display status of various I/O and to display and set manual output, simulated input/output, assign of signals and input of comment.
SETUP	The setting screen is used to set the system.
FILE	The file screen is used to read or store files. For example, files of program, system variable, and register.
USER	The user screen shows user messages when message instruction is executed.
SELECT	The program selection screen is used to list the programs. You can also create, copy and delete 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, string register, and pallet register.
STATUS	The status screen shows the system status.
4D GRAPHICS	This screen shows the 3D display of the robot and the current position data of the robot.
SYSTEM	The system screen is used to set system variables and mastering.
USER2	This screen displays messages output from KAREL programs.
BROWSER	This screen is used to browse Web pages on the network.

- Top menu

The top menu is used to select a screen. To display the top menu, press and hold the *i* key, then press the [MENU] key on the teach pendant.

By touching the icon on the touch panel (option), or entering the number at the upper left of the icon by the numeric key on the teach pendant, the screen corresponding to the selected icon is displayed. Up to 9 icons can be allocated in a top menu.

Up to 10 top menus can be defined. Each top menu can be switched by the function keys. The top menu for production, teaching, setting, and initial setup has been defined beforehand. (Refer to 11.4.5 Top menu setup.)

To disable a top menu, change the value of system variable $UI_CONFIG.$ENB_TOPMENU from TRUE to FALSE. This system variable can be used by system software 7DC2 (V8.20) series, version 01.or later.

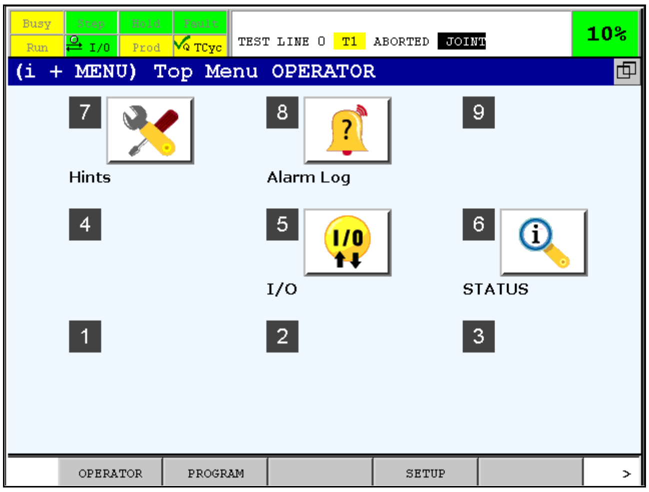


Fig. 2.3.1(h) Top menu

- 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.

FUNCTION 1	FUNCTION 2	FUNCTION 3
1 ABORT (ALL)	1 QUICK/FULL MENUS	1 REFRESH PANE
2 Disable FWD/BWD	2 SAVE	2
3 CAHNGE GROUP	3 PRINT SCREEN	3
4 TOGGLE SUB GROUP	4 PRINT	4
5 TOGGLE WRIST JOG	5	5
6	6 UNSIM ALL I/O	6
7 RELEASE WAIT	7	7 Diagnostic log
8	8 CYCLE POWER	8 Del Diag Log
9	9 ENABLE HMI MENUS	9
0 -- NEXT --	0 -- NEXT --	0 -- NEXT --
Page 1	**Page 2**	**Page 3**

Fig. 2.3.1(i) Function menu

Table 2.3.1 (j) Function menu

Item	Description
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. This item is displayed only when extended axis is setup.
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.

Item	Description
QUICK/FULL MENUS	QUICK/FULL MENUS toggles the menu between a usual screen menu and a quick menu.
SAVE	SAVE saves the data related to the current screen on external memory device.
PRINT SCREEN	PRINT SCREEN saves display data of the current screen to storage.
PRINT	PRINT prints program and system variables.
UNSIM ALL I/O	Cancels the simulated settings of all I/O signals.
CYCLE POWER	Makes a restart (power OFF/ON). (This item is not displayed if the controller is R-30*i*B Mate, R-30*i*B Mate Plus, R-30*i*B Compact Plus or R-30*i*B Mini Plus)
ENABLE HMI MENUS	Used to select whether to display the HMI menu when the MENU key is pressed.
REFRESH PANE	This item refreshes the pane in the active window.
Diagnostic log	This item saves the log data when problem occur on controller. This save operation should do before controller power off.
Del Diag Log	This item deletes the recorded diagnostic data.

Restart

It is now possible to make a restart (power off/on) from [FCTN] key (R-30*i*B Mate, R-30*i*B Mate Plus, R-30*i*B Compact Plus and R-30*i*B Mini Plus do not support restart by function menu).

- Condition

- The teach pendant is enabled.

- Step

1 Press the [FCTN] key.
2 Select "CYCLE POWER".
3 The screen below appears.

```
This will cycle power.
Are you sure?



     [ NO ]        YES
```

4 Select "YES" and press the [ENTER] key.

- 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. The screens that can be displayed depend on the application tool software. The screens that can be displayed in handling tool are as follows.

- ALARM / Alarm occurrence, Alarm history screen
- UTILITIES / Hint screen
- TEST CYCLE screen
- DATA / Register, Position Register screen
- MANUAL FCTNS screen
- I/O/ Digital I/O, Group I/O, Robot I/O screen
- STATUS/ Program, Axis, Version ID, Exec-hist, Memory screen
- Tool 1, Tool 2 screen
- USER, USER2 screen
- SETUP / Frames, Password screen

- 4D GRAPHICS/ 4D Display, Position Display screen
- BROWSER/ Browser, Panel setup screen

```
          QUICK 1
1 ALARM
2 UTILITY
3 TEST CYCLE
4 DATA
5 MANUAL FCTNS
6 I/O
7 STATUS
8
9 USER
0 -- NEXT --
```

Page 1

```
          QUICK 2
1 Tool1
2 Tool2
3 SETUP
4
5 4D GRAPHICS
6
7 USER2
8 BROWSER
9
0 -- NEXT --
```

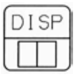
Page 2

Fig. 2.3.1(j) Quick menu

NOTE

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

Splitting screen

Pressing  key together with [SHIFT] key displays the following screen menu:

```
         DISPLAY  1
1 Single
2 Double
3 Triple
4 Status/Single
5 Single Wide
6 Double Horizontal
7 Triple Horizontal
8 Help/Diagnostics
9 EDIT<-->EDIT
0 -- NEXT --
```

Page 1

```
         DISPLAY  2
1 Menu History
2 User Views
3 Menu Favorites
4 Related Views
5 Maximize/Restore
6 Zoom
7
8
9
0 -- NEXXT --
```

Page 2

Fig. 2.3.1(k) Screen split menu

Table 2.3.1 (k) Description of the screen split menu

Item	Description
Single	Displays only one data item on the screen. The screen is not split.
Double	Splits the screen into right and left screens.
Triple	Splits the right screen into top and down screens and displays a total of three screens.
Status/Single	Splits the screen into right and left screens. The right screen is slightly larger than the left screen and the status sub-window with icons is displayed on the left screen.
Signal Wide	Up to 76 characters in a line and up to 20 lines can be displayed.
Double Horizontal	Splits the screen into top and down screens.
Triple Horizontal	Splits the top screen into right and left screens and displays a total of three screens.
EDIT<-->EDIT	When multiple edit screens are displayed, the program to edit is switched.
Menu History	The last displayed 8 menus are listed. The selected menu in the list can be displayed.
User Views	The list of the registered user view is displayed. The user view can be changed by selecting the registered user view from the list.

Item	Description
Menu Favorites	The list of the registered menu is displayed. The menu can be displayed by selecting the registered menu from the list.
Related Views	In case that the screen related to the current screen is registered, the related screen is displayed in the sub menu, the selected screen can be displayed.
Maximize/Restore	When the screen is split, the current screen is changed to full screen mode and return to the original screen mode.
Zoom	The character in the selected screen can be enlarged, and return to the original size.

Restrictions

- Two or three program edit screens can be displayed in the split windows at a time.
  But only the program opened in the left window is executable. To execute the program opened in the left window from the teach pendant, please select the left window as the current active window.
- Some screen, for example online touchup screen, can not be displayed at a same time in different windows.

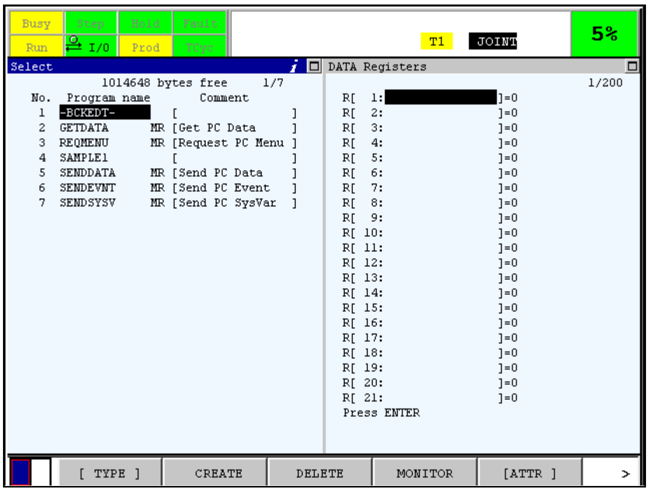


Fig. 2.3.1(l) Example of displaying double screens

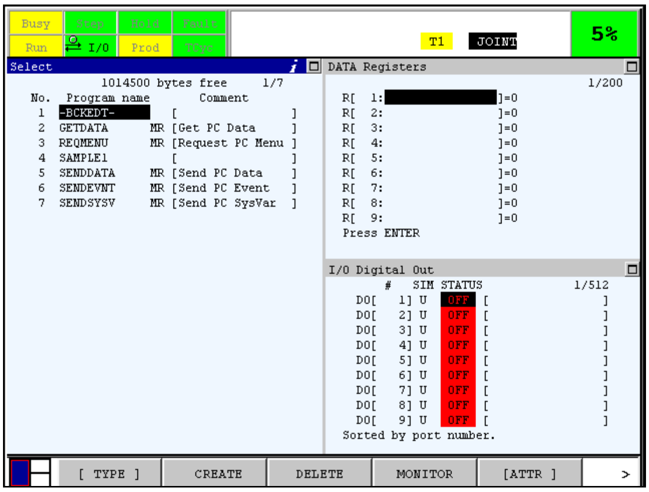


Fig. 2.3.1(m) Example of displaying triple screens

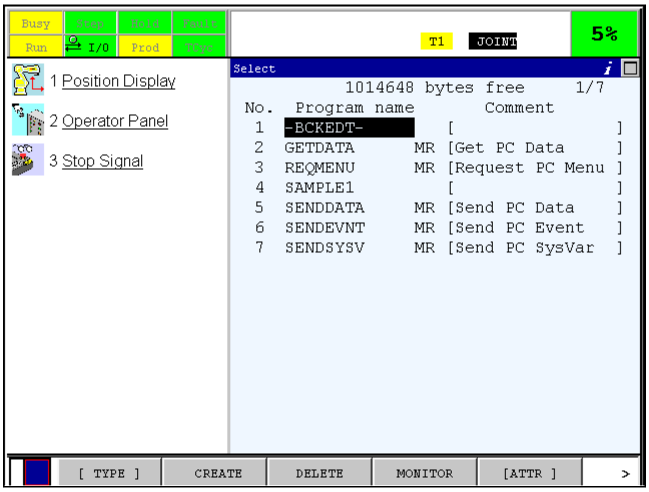


Fig. 2.3.1(n) Example of displaying the status/single screen

> **NOTE**
> When the amount of free memory in the DRAM memory is not enough, the screen may not be split and the alarm "TPIF-142 Menu limit exceeded" occurs. When the alarm "TPIF-142 Menu limit exceeded" occurs, please check the rest of temporary memory in the memory status display screen.

Changing the operation target screen

Pressing the 

key changes the operation target screen in turn. The title line of the screen which can be operated is displayed in blue and the frame of the screen is displayed in red.

ICON menu

During the pop-up menu invoked by pressing the [MENU], [DISP] or [FCTN] key is displayed on a screen, ICON menu is also displayed at the bottom of a screen for operation short cut. Favorite screen selection and user desired window configuration operation can be done quickly to select an ICON in the menu.

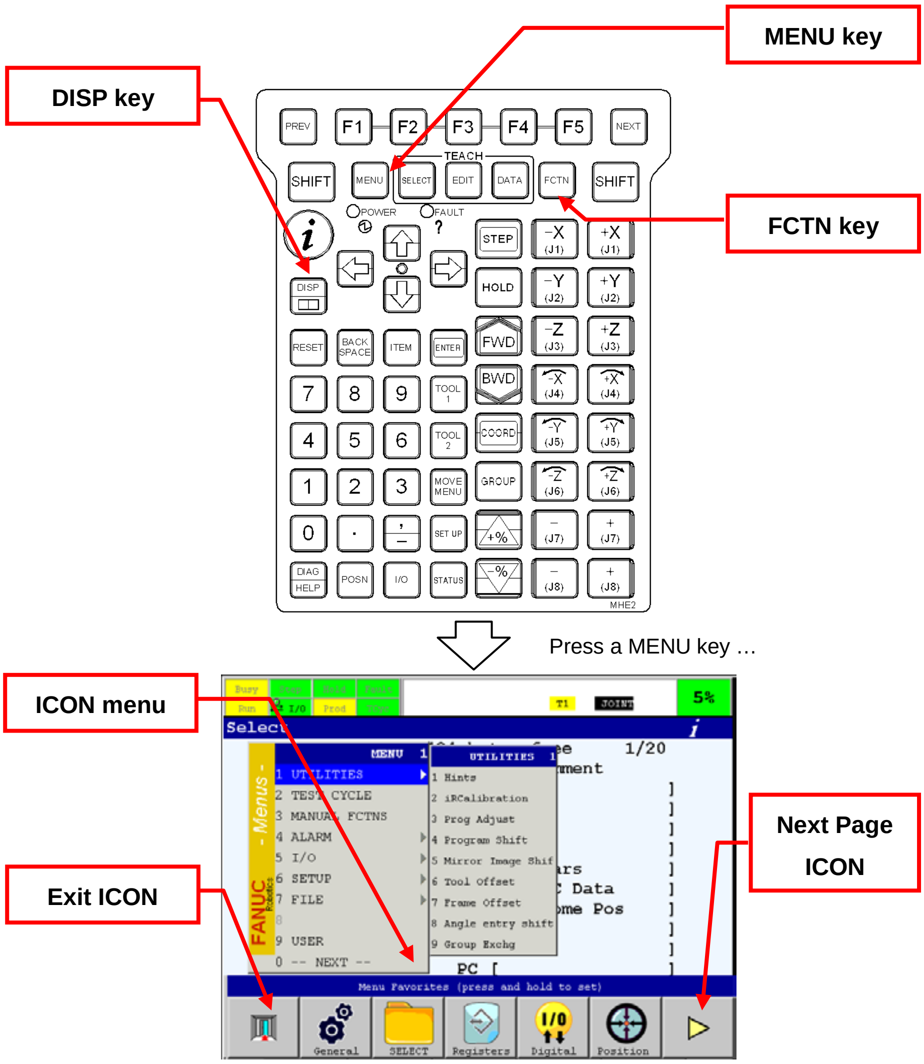


Fig. 2.3.1(o) ICON menu

If your teach pendant has a touch panel optional hardware, you can select an ICON by just only screen touching. If your teach pendant does not have a touch panel, you can select an ICON by pressing a function key, [PREV], or [NEXT] key located under an ICON.

The ICON at the right corner of the menu is the next page ICON to change the menu items to next page. The ICON at the left corner of the ICON menu is the exit ICON to escape from the ICON menu.

(A) ICON menu for favorite screen

When a MENU key is pressed, ICON menu for favorite screens will be displayed. By select an ICON of the menu, a favorite screen associated with the ICON will appear.

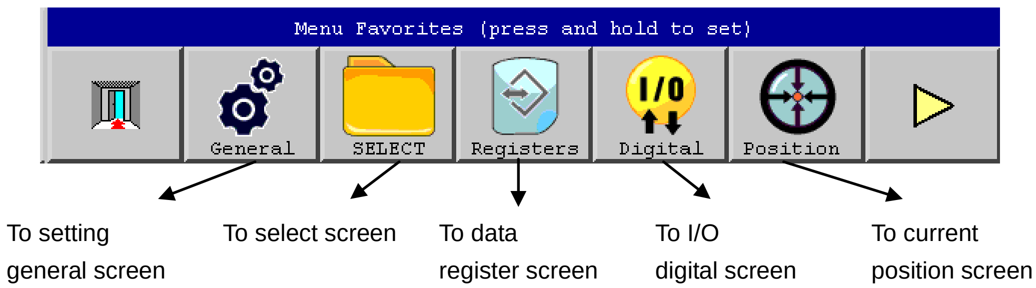


Fig. 2.3.1(p) ICON menu for favorite screen

At initial setting condition, no screen is registered into an ICON menu. To register a short cut to invoke a favorite screen, display your favorite screen and keep to press an ICON of the menu for about 4 seconds. Then the ICON for your favorite screen is registered into the pressed location in the menu. If your teach pendant does not have the touch panel optional hardware, please keep pressing a function key, PREV or [NEXT] key for 4 seconds. Up to 10 screens can be registered to your ICON menu.

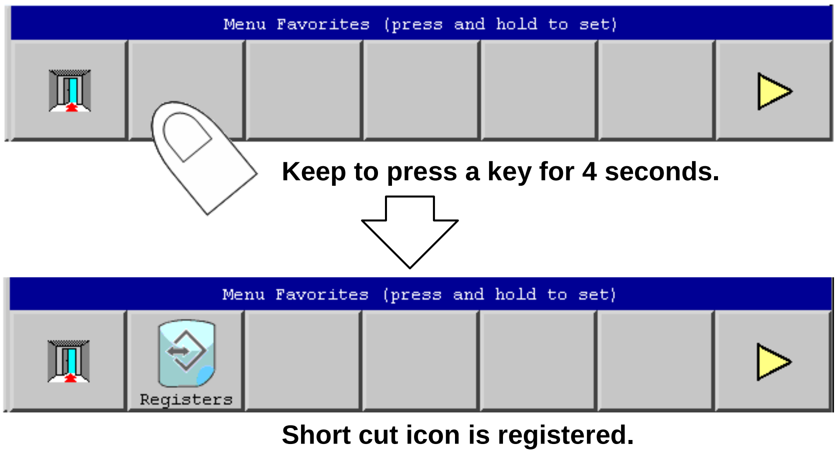


Fig. 2.3.1(q) Registration to ICON menu

In Arc Tool and Spot Tool+, some application related screens, that are used very often, have been registered into an ICON menu at FANUC shipping as the default setting.

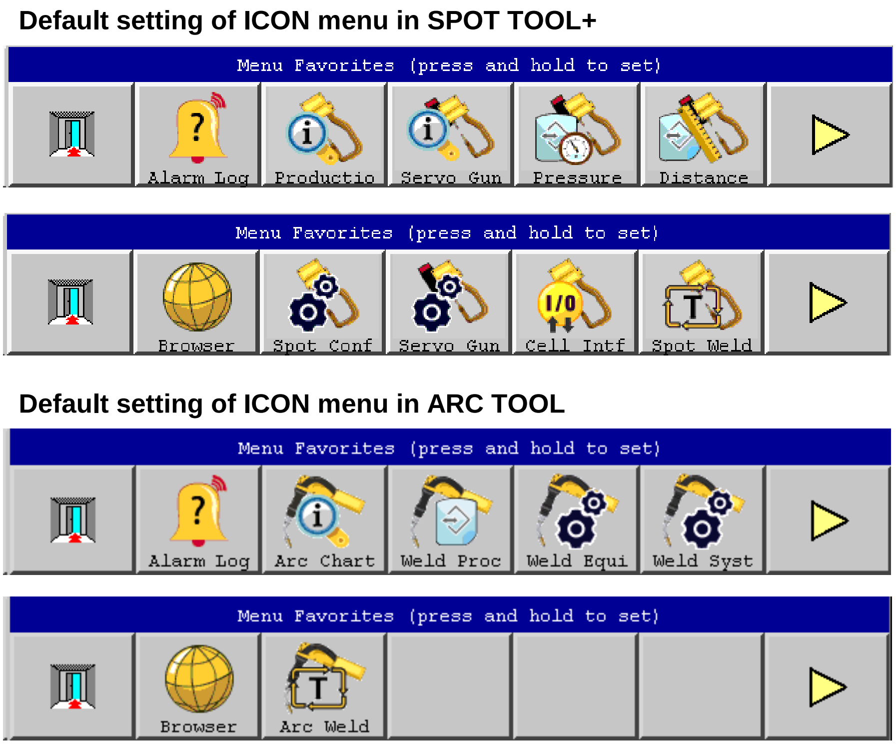


Fig. 2.3.1(r) Default setting of ICON menu corresponding application tool

(B) ICON menu for User Views

When [DISP] key and [SHIFT] key are pressed together, an ICON menu for User Views will be displayed. User Views means the user definable screen configuration and the window split mode and the selected screen in each window can be registered as a short cut ICON in the menu. When a short cut ICON is pressed, window is split and some screens are displayed in the windows according to the screen configuration automatically. Current window configuration is registered by pressing an ICON for about 4 seconds. If your teach pendant does not have a touch panel, use a function key, [PREV] or [NEXT] key. Up to 10 window configurations can be registered.

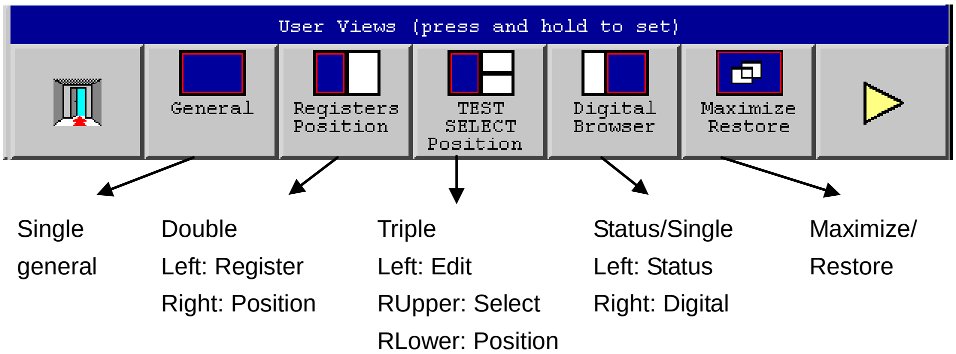


Fig. 2.3.1(s) User view ICON menu

A special ICON is registered as the default setting for window maximization operation at the far right of an ICON menu. When this icon is pressed, current selected window is maximized and displayed in full screen mode. And the ICON is pressed again, the maximized window is returned to previous size. If you don't need this maximization operation, you can overwrite this icon for another purpose.

> **NOTE**
> If you overwrite the maximization icon but want to recover it, please erase the fifth item of the ICON menu in *i*Pendant setup screen. Then the maximization icon will be recovered at the far right of your ICON menu. The maximization icon is always registered as the far right item and it can not be registered in other menu item location.

(C) ICON menu for sub-function

When [FCTN] key is pressed, the ICON menu for sub-function will be displayed. Contents of menu depend on applications. In SPOT TOOL+, short cut icons are registered for some functions, which are used very often for SPOT application.

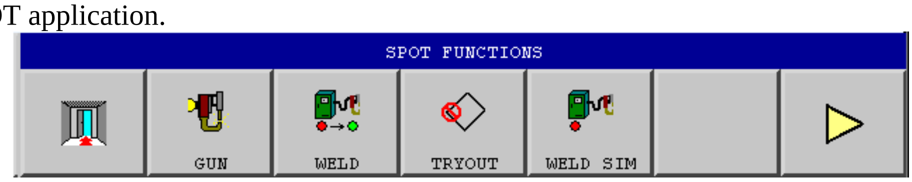


Fig. 2.3.1(t) ICON menu of FCTN menu of SPOT TOOL+

> **NOTE**
> - All short cut items in ICON menu for FCTN menu are decided statically and can not be customized. It is different from that of menu favorite and user menu.
> - ICON menu for sub-function is available depending on the application tool software.

(D) ICON menu for jog coordinate system

When [COORD] key is pressed while [SHIFT] key is held down, an ICON menu for jog coordinate system will be displayed at the bottom of screen. By selecting an ICON, the jog coordinate system can be changed quickly.

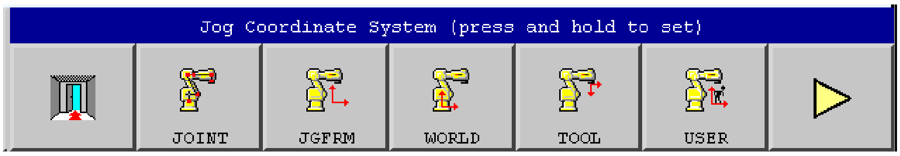


Fig. 2.3.1(u) ICON menu to change the jog coordinate system

ICON menu for jog coordinate system have been registered beforehand, and cannot be changed. ICON menu for jog coordinate system is enabled by default setting, but can be disabled in the *i*Pendant setup screen. Select *i*Pendant Setup in the menu displayed by MENU key to display the *i*Pendant setup screen, then press Menu Favorites Setup button in the *i*Pendant setup screen. The following screen will be displayed.

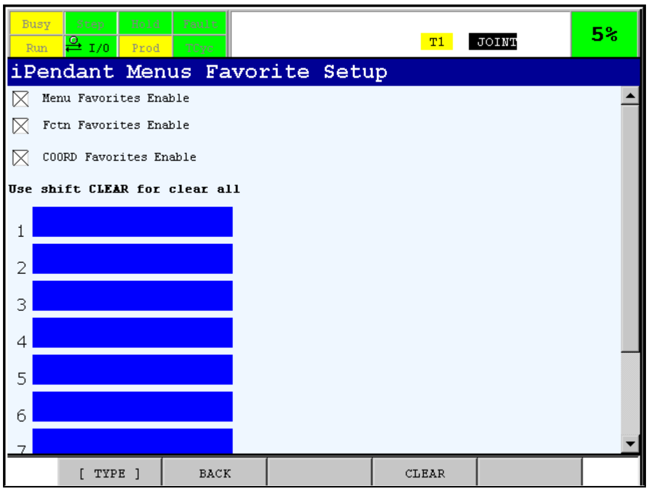


Fig. 2.3.1(v) *i*Pendant setup screen (menu favorite setup)

ICON menu for jog coordinate system can be enabled or disabled by checking the item "COORD Favorites Enable".

Software keyboard

You can use the software keyboard to input a character string. We would like to explain usage of software keyboard by showing how to display software keyboard to input comment of register.

Move cursor to the register that you want to input comment and press the [ENTER] key. The menu to input character is displayed around F5 key. Set the cursor to the item Options in the menu. Then display of F5 key is changed to KEYBOARD. Press the key.

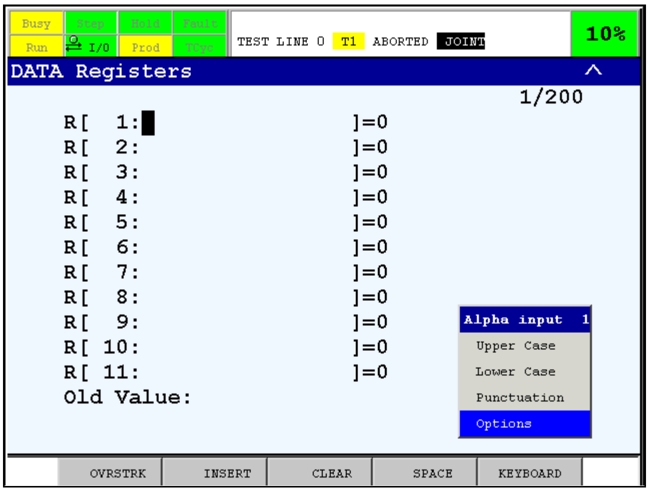


Fig. 2.3.1(w) Register screen

The software keyboard is displayed as follows.

NOTE
When entering into the text box using the software keyboard, make sure that the cursor is displayed in the text box, and then start up the software keyboard.

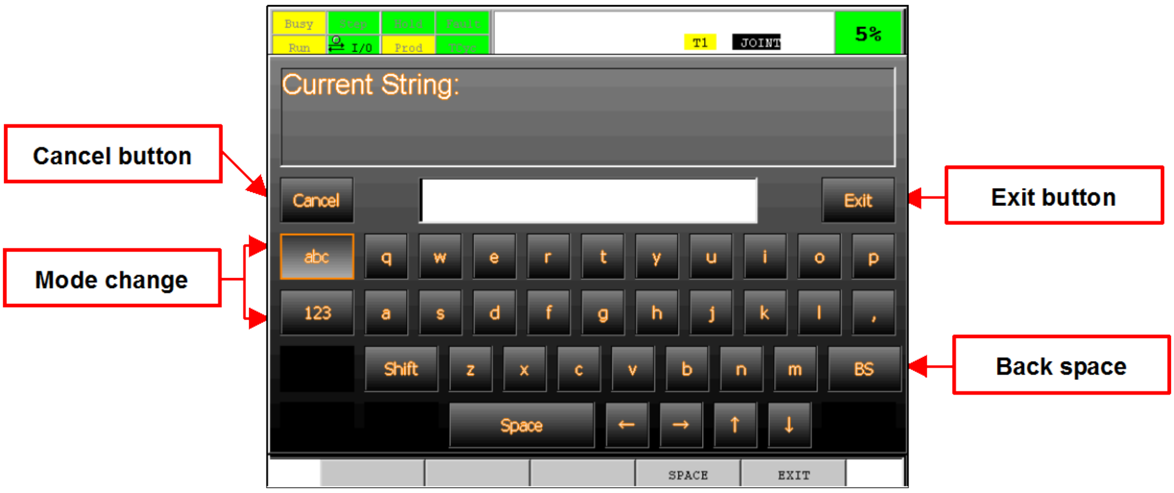


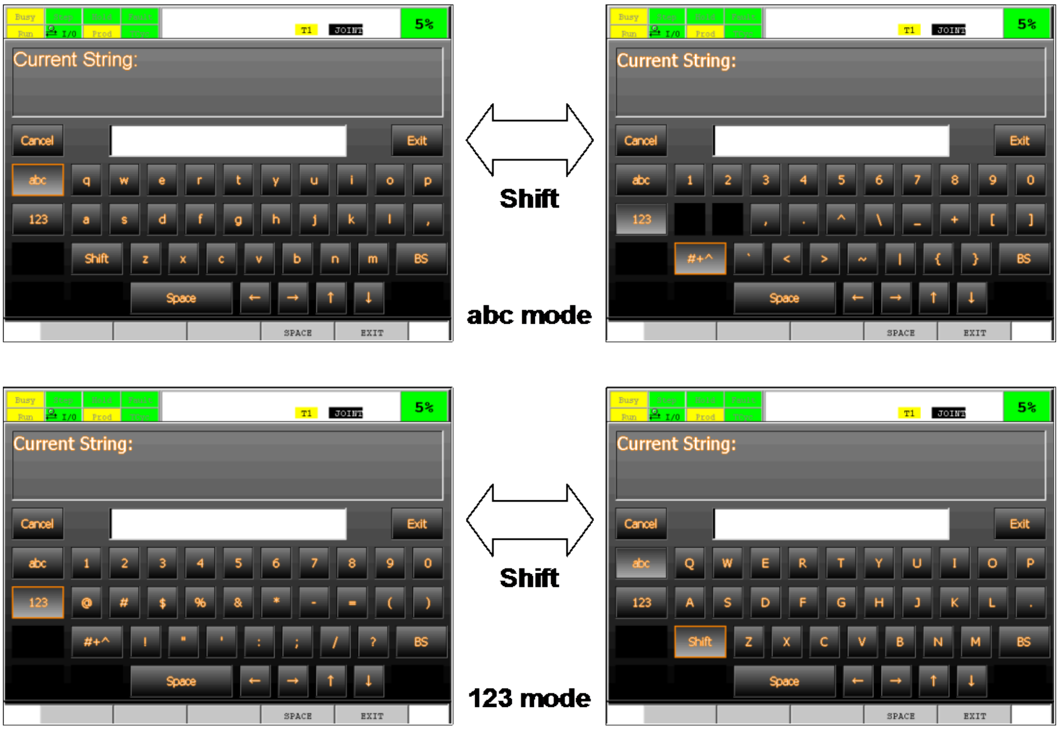
Fig. 2.3.1(x) software keyboard

In order to input character, move the cursor to the character that you want to input by the arrow key on the teach pendant, then press the [ENTER] key on the teach pendant. If the teach pendant has a touch panel optional hardware, you can input the character by touching the character on the touch panel.

In order to change the input mode, select ”abc” or “123” on the software keyboard. When “abc” is selected, the input mode becomes the alphabetic character input mode. And, when “123” is selected, the input mode becomes the numeric character and symbol input mode. When [Shift] key on the software keyboard is pressed, the character on the software keyboard is changed as follows.

You can input space by pressing the F4 SPACE.

After you finish the input, select the “Exit” button or press F5 EXIT to exit the software keyboard. If you want to cancel the input characters and return to the previous screen, select the “Cancel” button.



In some input fields, software keyboard is displayed by moving cursor to the field or pressing the [Enter] key. If only function key display is changed, press F1 KEYBOARD to display software keyboard.
You can change how to display keyboard by F2 AUTO and F3 MANUAL. In auto mode, software keyboard is displayed only by moving cursor to input field. In manual mode, you have to press the [Enter] key to display software keyboard.

Fig. 2.3.1(y) switching of auto/manual

Backlight automatic blanking

The teach pendant can turn off the backlight automatically for energy saving when any key is not pressed for a definite period of time.

The setting of the automatic blanking function can be changed by the following system variable. Restart (power off/on) is necessary to enable a change of the setting.
$UI_CONFIG.$BLNK_ENABLE
If TRUE, an automatic blanking function is valid. If FALSE, the function is invalid.
$UI_CONFIG.$BLNK_TIMER

When any key is not pressed in this time, turn off the backlight automatically. Unit is minute.
$UI_CONFIG.$BLNK_ALARM
If TRUE, turn off the backlight automatically regardless of an alarm. If FALSE, when an alarm occurs, disable the automatic blanking function This system variable ($UI_CONFIG.$BLNK_ALARM) can be used by system software 7DC2 (V8.20) series, version 01.or later.

NOTE
- When the teach pendant enable switch is turned to the enable position, the automatic blanking is not performed.
- During blanking, if any key is pressed, the displaying of the teach pendant will be recovered soon. FANUC recommends to press the [SHIFT] key if you want to recover the display.
- During blanking, any display is not visible on the screen. Don't judge the status of power cycle of your controller by whether display is visible or not on the screen. The teach pendant has a green LED indicator on a key sheet to show the status of a controller cycle power.

NetFront Browser

NetFront by ACCESS CO., LTD. is mounted in this product.

Part of the software of this product includes modules developed by Independent JPEG Group.

2.3.2 Operator Panel

The operator panel has buttons, switches, and connectors. Fig. 2.3.2 shows the operator panel on the cabinet.
The buttons on the operator panel/box can be used to start a program, release the alarm state, 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-232-C communication port and a USB communication port.
Table 2.3.2 (a) lists the switches on the operator's panel. Table 2.3.2 (b) lists the LEDs on the operator panel.
The standard operator panel of the robot controller does not have the power ON/OFF button. Execute the power on or off with the breaker of the controller.

Table 2.3.2 (a) Switches on the operator panel

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

Table 2.3.2 (b) LEDs on the operator panel

LED	Description
Alarm	Indicates the alarm state. Press the alarm release button to release the alarm state.
Power	Indicates that the power of the controller is ON.

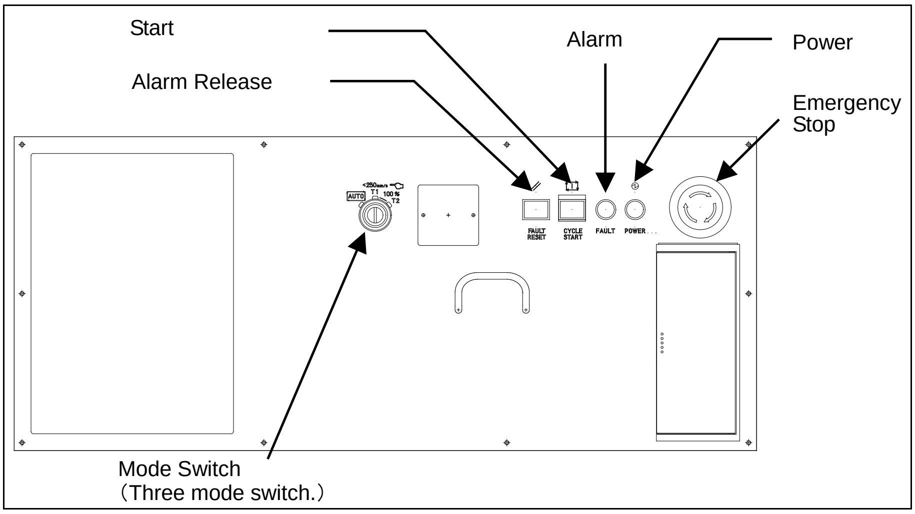


Fig. 2.3.2 (a) R-30*i*B Operator panel (standard)

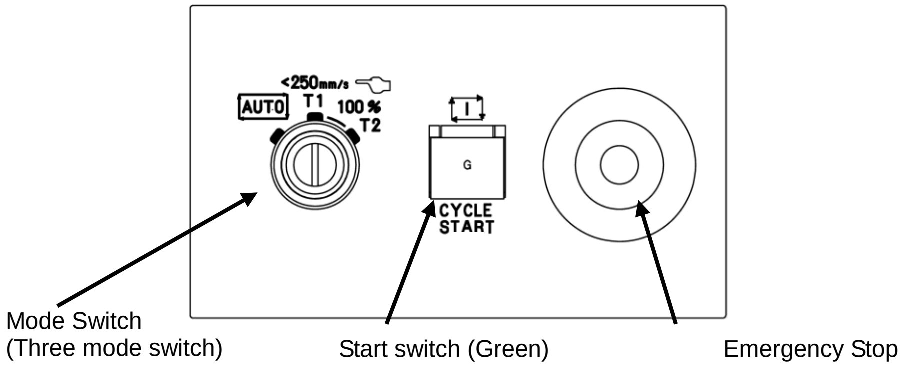


Fig. 2.3.2 (b) R-30*i*B Mate Operator panel

2.3.3 Remote Controller

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

2.3.4 CRT/KB

The CRT/KB is an optional operation unit. An external CRT/KB is connected to the controller via an RS-232-C cable.
The CRT/KB can be used to execute almost all teach pendant functions excluding those related to robot operation. Functions related to robot operation can only be executed using the teach pendant.

2.3.5 Communication

For communications, the following interfaces are provided (setting a communication port Section 8.2).

- Port 1 RS-232-C

- Port 2 RS-232-C (Port 2 is not available on R-30*i*B Mate, R-30*i*B Mate Plus, R-30*i*B Compact Plus and R-30*i*B Mini Plus controllers.)

2.3.6 Input/Output

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

The input/output signals include the following:
- Peripheral I/O (See Section 3.3.)
- Operator’s panel I/O (See Section 3.4.)
- Robot I/O (See Section 3.2.)
- Digital I/O (See Subsection 3.1.1.)
- Group I/O (See Subsection 3.1.2.)
- Analog I/O (See Subsection 3.1.3.)

The number of the I/O signals and their types depend on the hardware of the controller and the number of selected I/O modules and their types.
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, feed rate, and other factors. The Robot controller can control multiple axes, divided into multiple operation groups (multiple motion function). 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 feed rate override. When a motion instruction is used, the motion of the robot depends on the position data, motion format, positioning path, traveling speed, and feed rate override specified in the instruction.
One of four motion formats – "Linear", "Circular", "Circle Arc" 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" or "Circle Arc" is selected, the tool is moved along an arc connecting three specified points.
A positioning path can be selected from two options, "Fine" and "Cnt".

2.3.9 Emergency Stop Devices

This robot has following emergency stop devices.

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

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

2.3.10 Extended Axis

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

- Auxiliary axes
  This can move only at the joint motion.
- Integrated axes
  Controlled together with the robot during linear or circular or circle arc robot motion. Use these axes to perform linear or circular or circle arc robot motion.

Restrictions

- Limitation of the number of axes

The maximum number of axes in a group of independent axes or positioner is 4. You can add extended axes to this group, but the maximum number of total axes in this group including extended axes is also 4. You cannot add 5th and subsequent axes.
The maximum number of motors in a group is 9. If you use a robot which has 7 or more motors in itself, the maximum number of motors you can add in the same group would be less than 3.

- Limitation of motion range

The upper and lower limits of the motor (Pulsecoder) are ± 4085 rotations from the zero position. The motion range of the extended axis cannot exceed the upper and lower limits of the motor (Pulsecoder) rotations. The upper and lower limits of the motion range of the extended axis are determined as follows, using the reduction ratio of the mechanical parts driven by the motor.

Linear axis
Absolute value of upper and lower limits of motion range [mm] = 4085 x R_lin
R_lin: Axis movement [mm] per one motor rotation
Rotation axis
Absolute value of upper and lower limits of motion range [deg] = 4085 / R_rot x 360
R_rot: Number of motor rotation required for one axis rotation

Example 1: Linear axis, R_lin = 10
The motion range of the extended axis can be set only between -40850 [mm] and 40850 [mm].
Example 2: Case of rotation axis, R_rot = 100
The motion range of the extended axis can be set only between -14706 [deg] and 14706 [mm].

There is no such a limitation when using the Continuous Rotation function for a rotation axis. This axis can rotate in one direction infinitely. For the Continuous Rotation function, refer to the chapter of "Continuous Rotation function" in the FANUC Robot series Optional Function OPERATOR’S MANUAL (B-83284EN-2).

3 SETTING UP THE ROBOT SYSTEM

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

Contents of this chapter

3.1 I/O

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

For the safety signals like external emergency stop signal and safety fence signal, refer to the "FANUC Robot series R-30*i*B/R-30*i*B Plus CONTROLLER MAINTENANCE MANUAL" (B-83195EN) or the "FANUC Robot series R-30*i*B Mate/R-30*i*B Mate Plus CONTROLLER MAINTENANCE MANUAL" (B-83525EN).or the "FANUC Robot series R-30*i*B Mate Open Air MAINTENANCE MANUAL" (B-83555EN).

General-purpose I/O

The user can define the general-purpose I/O as required.
This group includes the following signals:

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

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

Specialized I/O

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

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

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

Configuring I/O

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

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

Rack

The rack indicates the kind of I/O module.

- 0 = Process I/O board, I/O link connection unit
- 1 to 15 = I/O Unit-MODEL A / B
- 32 = I/O link slave interface
- 33 = PMC (internal I/O assignment)
- 34 = Flags, Markers
- 35 = Always ON
- 36 = DCS Safe I/O (Non-safety function)
- 48 = R-30*i*B Mate main board (CRMA15, CRMA16)
- 66 = PROFIBUS-DP (master)
- 67 = PROFIBUS-DP (slave)
- 68 = FL-net (variety 1)
- 69 = FL-net status (variety 1)
- 75 = FIPIO (slave)
- 81 = DeviceNet (board 1)
- 82 = DeviceNet (board 2)
- 83 = DeviceNet (board 3)
- 84 = DeviceNet (board 4)
- 87 = RoboWeld
- 88 = Ethernet Global Data
- 89 = EthernetIP
- 90 = Arclink
- 91 = WTC welder
- 92 = CC-Link
- 93 = InterBus (master)
- 94 = InterBus (slave)
- 95 = InterBus (CMD mode)
- 96 = Modbus TCP
- 98 = InterBus slave only
- 99 = PROFINET I/O controller
- 100 = PROFINET I/O device
- 101 = Dual Channel PROFINET I/O controller
- 102 = Dual Channel PROFINET I/O device
- 103 = FL-net (variety 2)
- 104 = FL-net status (variety 2)
- 105 = CC-Link IE Field
- 106 = EtherCAT
- 107 = Tool Interface

Slot

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

- When the process I/O board or I/O link connection unit is used, the first connected device is Slot 1, the second is Slot 2 and others are numbered sequentially as this.

- When the I/O unit-MODEL A is used, the slot number on the base unit 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 is the unit number specified by the DIP switch in the basic unit.
- For I/O link slaves interface or R-30*i*B Mate main board (CRMA15, CRMA16), the number is always 1.
- For Flags, the number is always 1. And for Markers, the number is always 2.
- For Always ON, the number is always 1.
- For the other I/O modules, PMC (internal I/O assignment), DCS Safe I/O (Non-safety function), and from PROFIBUS-DP (master) to Tool Interface, refer to corresponding manuals.

Physical number

The physical number indicates the signal number in the I/O module. The physical number is expressed as follows.

- Digital input signal :
  The first signal is in1, the second signal is in2, and others are numbered sequentially as this.
- Digital output signal :
  The first signal is out1, the second signal is out2, and others are numbered sequentially as this.
- Analog input signal :
  The first signal is ain1, the second signal is ain2, and others are numbered sequentially as this.
- Analog output signal :
  The first signal is aout1, the second signal is aout2, and others are numbered sequentially as this.

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

- 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 and Operator’s panel I/O are always the same as the logic numbers. They cannot be redefined.

When I/O assignment is deleted and the power of the robot controller is turned on, the connected I/O devices are recognized and proper I/O assignment is automatically performed. The I/O assignment is called the **standard I/O assignment**.
The standard I/O assignment varies with the setting of "UOP auto assignment" in system config menu. (See Subsections 3.3, PERIPHERAL I/O)
The physical number and the standard I/O assignment of each hardware component that composes the I/O module are described below.

Process I/O board

The process I/O board is a I/O link slave device that has digital Input/Output signals and/or analog Input/Output signals. The type and number of signals varies with the type of process I/O board.

For connection of the process I/O board, refer to the “FANUC Robot series R-30*i*B/R-30*i*B Plus CONTROLLER MAINTENANCE MANUAL” (B-83195EN) or the “FANUC Robot series R-30*i*B Mate/R-30*i*B Mate Plus CONTROLLER MAINTENANCE MANUAL” (B-83525EN).

In the maintenance manuals, the logical signal name of the standard I/O assignment is specified as the signal name. The following table shows the physical number corresponded to the signal name specified in the maintenance manual and the standard I/O assignment in case of a process I/O board is connected.
The standard assignment of the process I/O board MA is different from the other process I/O board, because the number of signals is small.

Table 3.1 (a) Physical number and standard I/O assignment for digital Input/Output signals of Process I/O board (except process I/O board MA)

Physical number	R-30*i*B maintenance manual	R-30*i*B standard I/O asg. UOP auto asg.: Full	UOP auto asg.: Simple	UOP auto asg.: None, Full(Slave), Full(CRMA16), Simple(Slave), Simple(CRMA16)
in 1	*IMSTP	UI[1] *IMSTP	UI[2] *HOLD	DI[1]
in 2	*HOLD	UI[2] *HOLD	UI[5] RESET *1	DI[2]
in 3	*SFSPD	UI[3] *SFSPD	UI[6] START *2	DI[3]
in 4	CSTOPI	UI[4] CSTOPI	UI[8] ENBL	DI[4]
in 5	FAULT RESET	UI[5] FAULT RESET	UI[9] PNS1	DI[5]
in 6	START	UI[6] START	UI[10] PNS2	DI[6]
in 7	HOME	UI[7] HONE	UI[11] PNS3	DI[7]
in 8	ENBL	UI[8] ENBL	UI[12] PNS4	DI[8]
in 9	RSR1/PNS1	UI[9] RSR1/PNS1/STYLE1	DI[1]	DI[9]
in 10	RSR2/PNS2	UI[10] RSR2/PNS2/STYLE2	DI[2]	DI[10]
in 11	RSR3/PNS3	UI[11] RSR3/PNS3/STYLE3	DI[3]	DI[11]
in 12	RSR4/PNS4	UI[12] RSR4/PNS4/STYLE4	DI[4]	DI[12]
in 13	RSR5/PNS5	UI[13] RSR5/PNS5/STYLE5	DI[5]	DI[13]
in 14	RSR6/PNS6	UI[14] RSR6/PNS6/STYLE6	DI[6]	DI[14]
in 15	RSR7/PNS7	UI[15] RSR7/PNS7/STYLE7	DI[7]	DI[15]
in 16	RSR8/PNS8	UI[16] RSR8/PNS8/STYLE8	DI[8]	DI[16]
in 17	PNSTROBE	UI[17] PNSTROBE	DI[9]	DI[17]
in 18	PROD START	UI[18] PROD START	DI[10]	DI[18]
in 19	DI01	DI[1]	DI[11]	DI[19]
: in N :	DI(N-18)	DI[N-18]	DI[N-8]	DI[N]
out 1	CMDENBL	UO[1] CMDENBL	UO[1] CMDENBL	DO[1]
out 2	SYSRDY	UO[2] SYSRDY	UO[6] FAULT	DO[2]
out 3	PROGRUN	UO[3] PROGRUN	UO[9] BATALM	DO[3]
out 4	PAUSED	UO[4] PAUSED	UO[10] BUSY	DO[4]
out 5	HELD	UO[5] HELD	DO[1]	DO[5]
out 6	FAULT	UO[6] FAULT	DO[2]	DO[6]
out 7	ATPERCH	UO[7] ATPERCH	DO[3]	DO[7]
out 8	TPENBL	UO[8] TPENBL	DO[4]	DO[8]
out 9	BATALM	UO[9] BATALM	DO[5]	DO[9]
out 10	BUSY	UO[10] BUSY	DO[6]	DO[10]
out 11	ACK1/SNO1	UO[11] ACK1/SNO1	DO[7]	DO[11]
out 12	ACK2/SNO2	UO[12] ACK2/SNO2	DO[8]	DO[12]
out 13	ACK3/SNO3	UO[13] ACK3/SNO3	DO[9]	DO[13]
out 14	ACK4/SNO4	UO[14] ACK4/SNO4	DO[10]	DO[14]
out 15	ACK5/SNO5	UO[15] ACK5/SNO5	DO[11]	DO[15]
out 16	ACK6/SNO6	UO[16] ACK6/SNO6	DO[12]	DO[16]
out 17	ACK7/SNO7	UO[17] ACK7/SNO7	DO[13]	DO[17]
out 18	ACK8/SNO8	UO[18] ACK8/SNO8	DO[14]	DO[18]
out 19	SNACK	UO[19] SNACK	DO[15]	DO[19]
out 20	RESERVED	UO[20] Reserve	DO[16]	DO[20]
out 21	DO01	DO[1]	DO[17]	DO[21]
: out N :	DO(N-20)	DO[N-20]	DO[N-4]	DO[N]

*1 : in2 is also assigned to UI[4](CSTOPI).

*2 : in3 is also assigned to UI[17](PNSTROBE)

Table 3.1 (b) Physical number and standard I/O assignment for digital Input/Output signals of process I/O board MA

Physical number	R-30*i*B maintenance manual (R-30*i*B Mate is in parentheses)	R-30*i*B standard I/O asg. (R-30*i*B Mate is in parentheses)	
		UOP auto asg.: None Full Full(Slave) Full(CRMA16) Simple(Slave) Simple(CRMA16)	UOP auto asg.: Simple
in 1	DI01 (DI121)	DI[1] (DI[121])	UI[2] *HOLD
in 2	DI02 (DI122)	DI[2] (DI[122])	UI[5] RESET *1
in 3	DI03 (DI123)	DI[3] (DI[123])	UI[6] START *2
in 4	DI04 (DI124)	DI[4] (DI[124])	UI[8] ENBL
in 5	DI05 (DI125)	DI[5] (DI[125])	UI[9] PNS1
in 6	DI06 (DI126)	DI[6] (DI[126])	UI[10] PNS2
in 7	DI07 (DI127)	DI[7] (DI[127])	UI[11] PNS3
in 8	DI08 (DI128)	DI[8] (DI[128])	UI[12] PNS4
in 9	DI09 (DI129)	DI[9] (DI[129])	DI[1] (DI[121])
in 10	DI10 (DI130)	DI[10] (DI[130])	DI[2] (DI[122])
in 11	DI11 (DI131)	DI[11] (DI[131])	DI[3] (DI[123])
in 12	DI12 (DI132)	DI[12] (DI[132])	DI[4] (DI[124])
in 13	DI13 (DI133)	DI[13] (DI[133])	DI[5] (DI[125])
in 14	DI14 (DI134)	DI[14] (DI[134])	DI[6] (DI[126])
in 15	DI15 (DI135)	DI[15] (DI[135])	DI[7] (DI[127])
in 16	DI16 (DI136)	DI[16] (DI[136])	DI[8] (DI[128])
in 17	DI17 (DI137)	DI[17] (DI[137])	DI[9] (DI[129])
in 18	DI18 (DI138)	DI[18] (DI[138])	DI[10] (DI[130])
in 19	DI19 (DI139)	DI[19] (DI[139])	DI[11] (DI[131])
in 20	DI20 (DI140)	DI[20] (DI[140])	DI[12] (DI[132])
out 1	DO01 (DO121)	DO[1] (DO[121])	UO[1] CMDENBL
out 2	DO02 (DO122)	DO[2] (DO[122])	UO[6] FAULT
out 3	DO03 (DO123)	DO[3] (DO[123])	UO[9] BATALM
out 4	DO04 (DO124)	DO[4] (DO[124])	UO[10] BUSY
out 5	DO05 (DO125)	DO[5] (DO[125])	DO[1] (DO[121])
out 6	DO06 (DO126)	DO[6] (DO[126])	DO[2] (DO[122])
out 7	DO07 (DO127)	DO[7] (DO[127])	DO[3] (DO[123])
out 8	DO08 (DO128)	DO[8] (DO[128])	DO[4] (DO[124])
out 9	DO09 (DO129)	DO[9] (DO[129])	DO[5] (DO[125])
out 10	DO10 (DO130)	DO[10] (DO[130])	DO[6] (DO[126])
out 11	DO11 (DO131)	DO[11] (DO[131])	DO[7] (DO[127])
out 12	DO12 (DO132)	DO[12] (DO[132])	DO[8] (DO[128])
out 13	DO13 (DO133)	DO[13] (DO[133])	DO[9] (DO[129])
out 14	DO14 (DO134)	DO[14] (DO[134])	DO[10] (DO[130])
out 15	DO15 (DO135)	DO[15] (DO[135])	DO[11] (DO[131])
out 16	DO16 (DO136)	DO[16] (DO[136])	DO[12] (DO[132])

*1 : in2 is also assigned to UI[4](CSTOPI).
*2 : in3 is also assigned to UI[17](PNSTROBE)

Table 3.1 (c) Physical number and standard I/O assignment for analog Input/Output signals of process I/O board

Physical number	R-30*i*B / R-30*i*B Mate maintenance manual	R-30*i*B / R-30*i*B Mate standatrd I/O assignment
ain 1	ADCH1	AI[1]
ain 2	ADCH2	AI[2]
ain 3	ADCH3	AI[3]
ain 4	ADCH4	AI[4]
ain 5	ADCH5	AI[5]
ain 6	ADCH6	AI[6]
aout 1	DACH1	AO[1]
aout 2	DACH2	AO[2]

I/O Unit-MODEL A

I/O Unit-MODEL A is the I/O module which includes the plural modules. Plural modules can be connected. For details of the I/O Unit-MODEL A, refer to the “FANUC I/O Unit-MODEL A CONNECTION AND MAINTENANCE MANUAL“ (B-61813EN).

In “FANUC I/O Unit-MODEL A CONNECTION AND MAINTENANCE MANUAL“, the address that the signals are assigned in CNC is specified. The following table shows the physical number corresponded to the CNC address specified in the manual and the standard I/O assignment in case of a I/O module is connected. The standard I/O assignment is not changed by the setting of UOP auto assignment, because I/O Unit-MODEL A is not assigned to peripheral I/O by standard I/O assignment.

Table 3.1 (d) Physical number and standard I/O assignment for digital Input/Output signals of I/O Unit-MODEL A

Physical number	FANUC I/O Unit-MODEL A CONNECTION AND MAINTENANCE MANUAL		Standard I/O assignment	
	Address	Bit	R-30*i*B	R-30*i*B Mate
in 1	Xm	0	DI[1]	DI[121]
in 2	Xm	1	DI[2]	DI[122]
in 3	Xm	2	DI[3]	DI[123]
in 4	Xm	3	DI[4]	DI[124]
in 5	Xm	4	DI[5]	DI[125]
in 6	Xm	5	DI[6]	DI[126]
in 7	Xm	6	DI[7]	DI[127]
in 8	Xm	7	DI[8]	DI[128]
in 9	Xm+1	0	DI[9]	DI[129]
in 10	Xm+1	1	DI[10]	DI[130]
in 11	Xm+1	2	DI[11]	DI[131]
in 12	Xm+1	3	DI[12]	DI[132]
: in (8×A+B+1) :	Xm+A	B	DI[8×A+B+1]	DI[8×A+B+121]
out 1	Yn	0	DO[1]	DO[121]
out 2	Yn	1	DO[2]	DO[122]
out 3	Yn	2	DO[3]	DO[123]
out 4	Yn	3	DO[4]	DO[124]
out 5	Yn	4	DO[5]	DO[125]
out 6	Yn	5	DO[6]	DO[126]
out 7	Yn	6	DO[7]	DO[127]
out 8	Yn	7	DO[8]	DO[128]
out 9	Yn+1	0	DO[9]	DO[129]
out 10	Yn+1	1	DO[10]	DO[130]

Physical number	FANUC I/O Unit-MODEL A CONNECTION AND MAINTENANCE MANUAL		Standard I/O assignment	
	Address	Bit	R-30*i*B	R-30*i*B Mate
out 11	Yn+1	2	DO[11]	DO[131]
out 12	Yn+1	3	DO[12]	DO[132]
: out (8×A+B+1) :	Yn+A	B	DO[8×A+B+1]	DO[8×A+B+121]

Table 3.1 (e) Physical number and standard I/O assignment for analog Input/Output signals of I/O Unit-MODEL A

Physical number	FANUC I/O Unit-MODEL A CONNECTION AND MAINTENANCE MANUAL	R-30*i*B / R-30*i*B Mate standard I/O assignment
ain 1	Channel 0 of analog input module	AI[1]
ain 2	Channel 1 of analog input module	AI[2]
ain 3	Channel 2 of analog input module	AI[3]
ain 4	Channel 3 of analog input module	AI[4]
aout 1	Channel 0 of analog output module	AO[1]
aout 2	Channel 1 of analog output module	AO[2]

I/O Unit-MODEL B

The I/O Unit-MODEL B consists of an interface unit and one or more DI/DO units. 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.
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. (→ Section 3.5, I/O link screen)

In “FANUC I/O Unit-MODEL B CONNECTION MANUAL“, the address that the signals are assigned in CNC is specified. The physical number corresponded to the CNC address specified in the manual and the standard I/O assignment in case of a I/O module is connected are the same as the I/O Unit-MODEL A.

R-30*i*B Mate main board (CRMA15, CRMA16)

The R-30*i*B Mate controller is equipped with peripheral device control interfaces, which have 28 input points and 24 output points in total. By default, the signals of the peripheral device control interfaces are assigned to DI[101-120], DO[101-120], DI[81-88], and DO[81-84]. In the LR Handling Tool, the signals of R-30*i*B Mate I/O main board are assigned to peripheral I/O by standard I/O assignment, because "UOP auto assignment" is set to "Simple(CRMA16)".

For connection of R-30*i*B Mate I/O main board signals, refer to “FANUC Robot series R-30*i*B Mate/R-30*i*B Mate Plus CONTROLLER MAINTENANCE MANUAL” (B-83525EN).

In the maintenance manual, the logical signal name of the standard I/O assignment in LR Handling Tool is specified as the signal name. The following table shows the physical number corresponded to the signal name specified in the maintenance manual and the standard I/O assignment.

Table 3.1 (f) Physical number and standard I/O assignment for digital Input/Output signals of R-30*i*B Mate main board (CRMA15, CRMA16)

Physical number	R-30*i*B Mate maintenance manual	R-30*i*B Mate standard I/O assignment		
		UOP auto asg.: Simple(CRMA16)	UOP auto asg.: Full(CRMA16)	UOP auto asg.: None Full Full(Slave) Simple Simple(Slave)
in 1	DI101	DI[101]	UI[1] *IMSTP	DI[101]
in 2	DI102	DI[102]	UI[2] *HOLD	DI[102]
in 3	DI103	DI[103]	UI[3] *SFSPD	DI[103]
in 4	DI104	DI[104]	UI[4] CSTOPI	DI[104]
in 5	DI105	DI[105]	UI[5] FAULT RESET	DI[105]
in 6	DI106	DI[106]	UI[6] START	DI[106]
in 7	DI107	DI[107]	UI[7] HONE	DI[107]
in 8	DI108	DI[108]	UI[8] ENBL	DI[108]
in 9	DI109	DI[109]	UI[9] RSR1/PNS1/STYLE1	DI[109]
in 10	DI110	DI[110]	UI[10] RSR2/PNS2/STYLE2	DI[110]
in 11	DI111	DI[111]	UI[11] RSR3/PNS3/STYLE3	DI[111]
in 12	DI112	DI[112]	UI[12] RSR4/PNS4/STYLE4	DI[112]
in 13	DI113	DI[113]	UI[13] RSR5/PNS5/STYLE5	DI[113]
in 14	DI114	DI[114]	UI[14] RSR6/PNS6/STYLE6	DI[114]
in 15	DI115	DI[115]	UI[15] RSR7/PNS7/STYLE7	DI[115]
in 16	DI116	DI[116]	UI[16] RSR8/PNS8/STYLE8	DI[116]
in 17	DI117	DI[117]	UI[17] PNSTROBE	DI[117]
in 18	DI118	DI[118]	UI[18] PROD START	DI[118]
in 19	DI119	DI[119]	DI[119]	DI[119]
in 20	DI120	DI[120]	DI[120]	DI[120]
in 21	*HOLD	UI[2] *HOLD	DI[81]	DI[81]
in 22	RESET	UI[5] RESET *1	DI[82]	DI[82]
in 23	START	UI[6] START *2	DI[83]	DI[83]
in 24	ENBL	UI[8] ENBL	DI[84]	DI[84]
in 25	PNS1	UI[9] PNS1	DI[85]	DI[85]
in 26	PNS2	UI[10] PNS2	DI[86]	DI[86]
in 27	PNS3	UI[11] PNS3	DI[87]	DI[87]
in 28	PNS4	UI[12] PNS4	DI[88]	DI[88]
out 1	DO101	DO[101]	UO[1] CMDENBL	DO[101]
out 2	DO102	DO[102]	UO[2] SYSRDY	DO[102]
out 3	DO103	DO[103]	UO[3] PROGRUN	DO[103]
out 4	DO104	DO[104]	UO[4] PAUSED	DO[104]
out 5	DO105	DO[105]	UO[5] HELD	DO[105]
out 6	DO106	DO[106]	UO[6] FAULT	DO[106]
out 7	DO107	DO[107]	UO[7] ATPERCH	DO[107]
out 8	DO108	DO[108]	UO[8] TPENBL	DO[108]
out 9	DO109	DO[109]	UO[9] BATALM	DO[109]
out 10	DO110	DO[110]	UO[10] BUSY	DO[110]
out 11	DO111	DO[111]	UO[11] ACK1/SNO1	DO[111]
out 12	DO112	DO[112]	UO[12] ACK2/SNO2	DO[112]
out 13	DO113	DO[113]	UO[13] ACK3/SNO3	DO[113]
out 14	DO114	DO[114]	UO[14] ACK4/SNO4	DO[114]
out 15	DO115	DO[115]	UO[15] ACK5/SNO5	DO[115]
out 16	DO116	DO[116]	UO[16] ACK6/SNO6	DO[116]
out 17	DO117	DO[117]	UO[17] ACK7/SNO7	DO[117]
out 18	DO118	DO[118]	UO[18] ACK8/SNO8	DO[118]
out 19	DO119	DO[119]	UO[19] SNACK	DO[119]

Physical number	R-30*i*B Mate maintenance manual	R-30*i*B Mate standard I/O assignment UOP auto asg.: Simple(CRMA16)	UOP auto asg.: Full(CRMA16)	UOP auto asg.: None Full Full(Slave) Simple Simple(Slave)
out 20	DO120	DO[120]	UO[20] Reserve	DO[120]
out 21	CMDENBL	UO[1] CMDENBL	DO[81]	DO[81]
out 22	FAULT	UO[6] FAULT	DO[82]	DO[82]
out 23	BATALM	UO[9] BATALM	DO[83]	DO[83]
out 24	BUSY	UO[10] BUSY	DO[84]	DO[84]

*1 : in22 is also assigned to UI[4](CSTOPI).
*2 : in23 is also assigned to UI[17](PNSTROBE)

R-30*i*B Compact Plus and R-30*i*B Mini Plus main board (JRM18)

The R-30*i*B Compact Plus controller and R-30*i*B Mini Plus controller are equipped with peripheral device control interfaces, which have 20 input points and 16 output points in total. By default, the signals of the peripheral device control interfaces are assigned to DI[101-112], DO[101-112], DI[81-88], and DO[81-84]. In the LR Handling Tool, the signals of R-30*i*B Compact Plus and R-30*i*B Mini Plus I/O main boards are assigned to peripheral I/O by standard I/O assignment, because "UOP auto assignment" is set to "Simple(JRM18)".

For connection of R-30*i*B Compact Plus and I/O main board signals, refer to “FANUC Robot series R-30*i*B Compact Plus CONTROLLER MAINTENANCE MANUAL” (B-84035EN). For connection of R-30*i*B Mini Plus I/O main board signals, refer to “FANUC Robot series R-30*i*B Mini Plus CONTROLLER MAINTENANCE MANUAL” (B-84175EN).

In the maintenance manual, the logical signal name of the standard I/O assignment in LR Handling Tool is specified as the signal name. The following table shows the physical number corresponded to the signal name specified in the maintenance manual and the standard I/O assignment.

Table 3.1 (g) Physical number and standard I/O assignment for digital Input/Output signals of R-30*i*B Compact Plus and R-30*i*B Mini Plus main boards (JRM18)

Physical number	R-30*i*B Compact Plus, R-30*i*B Mini Plus maintenance manual	R-30*i*B Compact Plus, R-30*i*B Mini Plus standard I/O assignment UOP auto asg.: Simple(JRM18)	UOP auto asg.: None Full Full(Slave) Simple Simple(Slave)
in 1	DI101	DI[101]	DI[101]
in 2	DI102	DI[102]	DI[102]
in 3	DI103	DI[103]	DI[103]
in 4	DI104	DI[104]	DI[104]
in 5	DI105	DI[105]	DI[105]
in 6	DI106	DI[106]	DI[106]
in 7	DI107	DI[107]	DI[107]
in 8	DI108	DI[108]	DI[108]
in 9	DI109	DI[109]	DI[109]
in 10	DI110	DI[110]	DI[110]
in 11	DI111	DI[111]	DI[111]
in 12	DI112	DI[112]	DI[112]
in 13	*HOLD	UI[2] *HOLD	DI[81]

Physical number	R-30*i*B Compact Plus, R-30*i*B Mini Plus maintenance manual	R-30*i*B Compact Plus, R-30*i*B Mini Plus standard I/O assignment	
		UOP auto asg.: Simple(JRM18)	UOP auto asg.: None Full Full(Slave) Simple Simple(Slave)
in 14	RESET	UI[5] RESET *1	DI[82]
in 15	START	UI[6] START *2	DI[83]
in 16	ENBL	UI[8] ENBL	DI[84]
in 17	PNS1	UI[9] PNS1	DI[85]
in 18	PNS2	UI[10] PNS2	DI[86]
in 19	PNS3	UI[11] PNS3	DI[87]
in 20	PNS4	UI[12] PNS4	DI[88]
out 1	DO101	DO[101]	DO[101]
out 2	DO102	DO[102]	DO[102]
out 3	DO103	DO[103]	DO[103]
out 4	DO104	DO[104]	DO[104]
out 5	DO105	DO[105]	DO[105]
out 6	DO106	DO[106]	DO[106]
out 7	DO107	DO[107]	DO[107]
out 8	DO108	DO[108]	DO[108]
out 9	DO109	DO[109]	DO[109]
out 10	DO110	DO[110]	DO[110]
out 11	DO111	DO[111]	DO[111]
out 12	DO112	DO[112]	DO[112]
out 13	CMDENBL	UO[1] CMDENBL	DO[81]
out 14	FAULT	UO[6] FAULT	DO[82]
out 15	BATALM	UO[9] BATALM	DO[83]
out 16	BUSY	UO[10] BUSY	DO[84]

*1 : in14 is also assigned to UI[4](CSTOPI).
*2 : in15 is also assigned to UI[17](PNSTROBE)

R-30*i*B Mate I/O link (master/slave)

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

I/O link slave mode

The robot controller operates as an I/O link slave device and connects to an I/O link master device such as the CNC. The I/O information is transferred to or from the CNC by I/O link slave interface which have 72 input points and 68 output points.

I/O link master mode

The robot controller operates as an I/O link master device and connects to an I/O link slave device. To use process I/O board, I/O Unit-MODEL A / B, I/O link connection unit, the I/O link master mode needs to be selected. For physical number and standard I/O assignment of the connected slave device, refer to each hardware item in "3.1 I/O".

In default setting, I/O link is set as follows.

- LR tool : I/O link slave mode
- LR handling tool : I/O link master mode.

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

To switch between the I/O link modes unrelated to the setting of UOP auto assignment, change system variable $IOMASTER and cycle power of the controller.

- $IOMASTER=0: I/O link slave mode
- $IOMASTER=1: I/O link master mode

For connection of R-30*i*B Mate I/O link, refer to "FANUC Robot series R-30*i*B Mate/R-30*i*B Mate Plus CONTROLLER MAINTENANCE MANUAL" (B-83525EN) or "FANUC Robot series R-30*i*B Mate Open Air MAINTENANCE MANUAL" (B-83555EN).

The following table shows the physical number corresponded to the address of the connected CNC and the standard I/O assignment in case of I/O link slave interface is used.

Table 3.1 (h) Physical number and standard I/O assignment for digital Input/Output signals of R-30*i*B Mate I/O link slave interface

Physical number	Address of CNC		R-30*i*B Mate standard I/O assignment		
			UOP auto asg.: Simple(Slave)	UOP auto asg.: Full(Slave)	UOP auto asg.: None Full Full(CRMA16) Simple Simple(CRMA16)
	Address	Bit			
in 1	Yn	0	UI[2] *HOLD	UI[1] *IMSTP	DI[1]
in 2	Yn	1	UI[5] RESET *1	UI[2] *HOLD	DI[2]
in 3	Yn	2	UI[6] START *2	UI[3] *SFSPD	DI[3]
in 4	Yn	3	UI[8] ENBL	UI[4] CSTOPI	DI[4]
in 5	Yn	4	UI[9] PNS1	UI[5] FAULT RESET	DI[5]
in 6	Yn	5	UI[10] PNS2	UI[6] START	DI[6]
in 7	Yn	6	UI[11] PNS3	UI[7] HONE	DI[7]
in 8	Yn	7	UI[12] PNS4	UI[8] ENBL	DI[8]
in 9	Yn+1	0	DI[1]	UI[9] RSR1/PNS1/STYLE1	DI[9]
in 10	Yn+1	1	DI[2]	UI[10] RSR2/PNS2/STYLE2	DI[10]
in 11	Yn+1	2	DI[3]	UI[11] RSR3/PNS3/STYLE3	DI[11]
in 12	Yn+1	3	DI[4]	UI[12] RSR4/PNS4/STYLE4	DI[12]
in 13	Yn+1	4	DI[5]	UI[13] RSR5/PNS5/STYLE5	DI[13]
in 14	Yn+1	5	DI[6]	UI[14] RSR6/PNS6/STYLE6	DI[14]
in 15	Yn+1	6	DI[7]	UI[15] RSR7/PNS7/STYLE7	DI[15]
in 16	Yn+1	7	DI[8]	UI[16] RSR8/PNS8/STYLE8	DI[16]
in 17	Yn+2	0	DI[9]	UI[17] PNSTROBE	DI[17]
in 18	Yn+2	1	DI[10]	UI[18] PROD START	DI[18]
in 19	Yn+2	2	DI[11]	DI[1]	DI[19]
: in (8×A+B+1) :	Yn+A	B	DI[8×A+B-7]	DI[8×A+B-17]	DI[8×A+B+1]
in 72	Yn+8	7	DI[64]	DI[54]	DI[72]
out 1	Xm	0	UO[1] CMDENBL	UO[1] CMDENBL	DO[1]
out 2	Xm	1	UO[6] FAULT	UO[2] SYSRDY	DO[2]
out 3	Xm	2	UO[9] BATALM	UO[3] PROGRUN	DO[3]
out 4	Xm	3	UO[10] BUSY	UO[4] PAUSED	DO[4]
out 5	Xm	4	DO[1]	UO[5] HELD	DO[5]
out 6	Xm	5	DO[2]	UO[6] FAULT	DO[6]
out 7	Xm	6	DO[3]	UO[7] ATPERCH	DO[7]
out 8	Xm	7	DO[4]	UO[8] TPENBL	DO[8]
out 9	Xm+1	0	DO[5]	UO[9] BATALM	DO[9]
out 10	Xm+1	1	DO[6]	UO[10] BUSY	DO[10]
out 11	Xm+1	2	DO[7]	UO[11] ACK1/SNO1	DO[11]

Physical number	Address of CNC		R-30*i*B Mate standard I/O assignment		
			UOP auto asg.: Simple(Slave)	**UOP auto asg.: Full(Slave)**	**UOP auto asg.: None Full Full(CRMA16) Simple Simple(CRMA16)**
	Address	**Bit**			
out 12	Xm+1	3	DO[8]	UO[12] ACK2/SNO2	DO[12]
out 13	Xm+1	4	DO[9]	UO[13] ACK3/SNO3	DO[13]
out 14	Xm+1	5	DO[10]	UO[14] ACK4/SNO4	DO[14]
out 15	Xm+1	6	DO[11]	UO[15] ACK5/SNO5	DO[15]
out 16	Xm+1	7	DO[12]	UO[16] ACK6/SNO6	DO[16]
out 17	Xm+2	0	DO[13]	UO[17] ACK7/SNO7	DO[17]
out 18	Xm+2	1	DO[14]	UO[18] ACK8/SNO8	DO[18]
out 19	Xm+2	2	DO[15]	UO[19] SNACK	DO[19]
out 20	Xm+2	3	DO[16]	UO[20] Reserve	DO[20]
out 21	Xm+2	4	DO[17]	DO[1]	DO[21]
: out (8×A+B+1) :	Xm+A	B	DO[8×A+B-3]	DO[8×A+B-19]	DO[8×A+B+1]
out 68	Xm+8	3	DO[64]	DO[48]	DO[68]

*1 : in2 is also assigned to UI[4](CSTOPI).
*2 : in3 is also assigned to UI[17](PNSTROBE)

NOTE

A hardware that supports I/O Link *i* slave has I/O Link *i* slave function for 2 groups. (That is for supporting the communication of dual signal for safety signals as slave. Refer to "FANUC Robot series R-30*i*B/R-30*i*B Mate/R-30*i*B Plus/R-30*i*B Mate Plus/R-30*i*B Compact Plus/R-30*i*B Mini Plus CONTROLLER Dual Check Safety Function OPERATOR'S MANUAL"(B-83184EN) and see Section "I/O LINK *i* SLAVE FUNCTION" for more information.)

If you connect the hardware as slave for CNC, please assign entire 2 groups to CNC. And for 2nd group, please assign 1 byte to free address of X or Y. If no byte is assigned for 2nd group, CNC posts the alarm "ER97 I/O Link FAILURE", and CNC doesn't communicate. (If you use dual signal for safety signals as slave, CNC doesn't post the alarm because entire 2 groups are assigned.)

R-30*i*B I/O link slave

The R-30*i*B controller operates as an I/O link slave device and connects to an I/O link master device such as the CNC by the setting of the following system variable.

- $IOSLAVE.$INPUT_N : Number of I/O link slave input signals.
- $IOSLAVE.$OUTPUT_N : Number of I/O link slave output signals.

By default, the number of signals is set to 0, and I/O link slave is not available. The maximum number of signals is 256 point input and 256 point output.

The R-30*i*B controller can use both I/O link master and slave at the same time.

For connection of R-30*i*B I/O link slave, refer to "FANUC Robot series R-30*i*B/R-30*i*B Mate Plus CONTROLLER MAINTENANCE MANUAL" (B-83195EN).
The physical number corresponded to the address of the connected CNC and the standard I/O assignment is the same as R-30*i*B Mate. Refer to Table 3.1 (g).

NOTE

A hardware that supports I/O Link *i* slave has I/O Link *i* slave function for 2 groups. (That is for supporting the communication of dual signal for safety signals as slave. Refer to “FANUC Robot series R-30*i*B/R-30*i*B/R-30*i*B Plus Mate/R-30*i*B Compact Plus/R-30*i*B Mini Plus CONTROLLER Dual Check Safety Function OPERATOR'S MANUAL”(B-83184EN) and see Section “I/O LINK *i* SLAVE FUNCTION” for more information.)

If you connect the hardware as slave for CNC, please assign entire 2 groups to CNC. And for 2nd group, please assign 1 byte to free address of X or Y. If no byte is assigned for 2nd group, CNC posts the alarm "ER97 I/O Link FAILURE", and CNC doesn’t communicate. (If you use dual signal for safety signals as slave, CNC doesn’t post the alarm because entire 2 groups are assigned.)

I/O link connection unit

The robot controller of I/O link master mode can connect to an I/O link master device such as the CNC by using I/O link connection unit. The maximum 256 input points and 256 output points I/O information is transferred to or from the CNC.
When the I/O link connection unit is used, the setting is needed on I/O link screen. (→ Section 3.5, ”I/O link screen”)

For connection of I/O link connection unit, refer to “FANUC Robot series R-30*i*B Mate/R-30*i*B Mate Plus CONTROLLER MAINTENANCE MANUAL” (B-83525EN).
The following table shows the physical number corresponded to the address of the connected CNC and the standard I/O assignment in case of I/O link connection unit is used.

Table 3.1 (i) Physical number and standard I/O assignment for digital Input/Output signals of I/O link connection unit

Physical number	Address of CNC		R-30*i*B standatd I/O asg. (R-30*i*B Mate is in parentheses)		
			UOP auto asg.: Simple	UOP auto asg.: Full	UOP auto asg.: None Full(Slave) Full(CRMA16) Simple(Slave) Simple(CRMA16)
	Address	Bit			
in 1	Yn	0	UI[2] *HOLD	UI[1] *IMSTP	DI[1] (DI[121])
in 2	Yn	1	UI[5] RESET *1	UI[2] *HOLD	DI[2] (DI[122])
in 3	Yn	2	UI[6] START *2	UI[3] *SFSPD	DI[3] (DI[123])
in 4	Yn	3	UI[8] ENBL	UI[4] CSTOPI	DI[4] (DI[124])
in 5	Yn	4	UI[9] PNS1	UI[5] FAULT RESET	DI[5] (DI[125])
in 6	Yn	5	UI[10] PNS2	UI[6] START	DI[6] (DI[126])
in 7	Yn	6	UI[11] PNS3	UI[7] HONE	DI[7] (DI[127])
in 8	Yn	7	UI[12] PNS4	UI[8] ENBL	DI[8] (DI[128])
in 9	Yn+1	0	DI[1] (DI[121])	UI[9] RSR1/PNS1/STYLE1	DI[9] (DI[129])
in 10	Yn+1	1	DI[2] (DI[122])	UI[10] RSR2/PNS2/STYLE2	DI[10] (DI[130])
in 11	Yn+1	2	DI[3] (DI[123])	UI[11] RSR3/PNS3/STYLE3	DI[11] (DI[131])
in 12	Yn+1	3	DI[4] (DI[124])	UI[12] RSR4/PNS4/STYLE4	DI[12] (DI[132])
in 13	Yn+1	4	DI[5] (DI[125])	UI[13] RSR5/PNS5/STYLE5	DI[13] (DI[133])
in 14	Yn+1	5	DI[6] (DI[126])	UI[14] RSR6/PNS6/STYLE6	DI[14] (DI[134])
in 15	Yn+1	6	DI[7] (DI[127])	UI[15] RSR7/PNS7/STYLE7	DI[15] (DI[135])
in 16	Yn+1	7	DI[8] (DI[128])	UI[16] RSR8/PNS8/STYLE8	DI[16] (DI[136])
in 17	Yn+2	0	DI[9] (DI[129])	UI[17] PNSTROBE	DI[17] (DI[137])
in 18	Yn+2	1	DI[10] (DI[130])	UI[18] PROD START	DI[18] (DI[138])

Physical number	Address of CNC		R-30*i*B standatd I/O asg. (R-30*i*B Mate is in parentheses)		
	Address	Bit	UOP auto asg.: Simple	UOP auto asg.: Full	UOP auto asg.: None Full(Slave) Full(CRMA16) Simple(Slave) Simple(CRMA16)
in 19	Yn+2	2	DI[11] (DI[131])	DI[1] (DI[121])	DI[19] (DI[139])
: in (8×A+B+1) :	Yn+A	B	DI[8×A+B-7] (DI[8×A+B+113])	DI[8×A+B-17] (DI[8×A+B+103])	DI[8×A+B+1] (DI[8×A+B+121])
out 1	Xm	0	UO[1] CMDENBL	UO[1] CMDENBL	DO[1] (DO[121])
out 2	Xm	1	UO[6] FAULT	UO[2] SYSRDY	DO[2] (DO[122])
out 3	Xm	2	UO[9] BATALM	UO[3] PROGRUN	DO[3] (DO[123])
out 4	Xm	3	UO[10] BUSY	UO[4] PAUSED	DO[4] (DO[124])
out 5	Xm	4	DO[1] (DO[121])	UO[5] HELD	DO[5] (DO[125])
out 6	Xm	5	DO[2] (DO[122])	UO[6] FAULT	DO[6] (DO[126])
out 7	Xm	6	DO[3] (DO[123])	UO[7] ATPERCH	DO[7] (DO[127])
out 8	Xm	7	DO[4] (DO[124])	UO[8] TPENBL	DO[8] (DO[128])
out 9	Xm+1	0	DO[5] (DO[125])	UO[9] BATALM	DO[9] (DO[129])
out 10	Xm+1	1	DO[6] (DO[126])	UO[10] BUSY	DO[10] (DO[130])
out 11	Xm+1	2	DO[7] (DO[127])	UO[11] ACK1/SNO1	DO[11] (DO[131])
out 12	Xm+1	3	DO[8] (DO[128])	UO[12] ACK2/SNO2	DO[12] (DO[132])
out 13	Xm+1	4	DO[9] (DO[129])	UO[13] ACK3/SNO3	DO[13] (DO[133])
out 14	Xm+1	5	DO[10] (DO[130])	UO[14] ACK4/SNO4	DO[14] (DO[134])
out 15	Xm+1	6	DO[11] (DO[131])	UO[15] ACK5/SNO5	DO[15] (DO[135])
out 16	Xm+1	7	DO[12] (DO[132])	UO[16] ACK6/SNO6	DO[16] (DO[136])
out 17	Xm+2	0	DO[13] (DO[133])	UO[17] ACK7/SNO7	DO[17] (DO[137])
out 18	Xm+2	1	DO[14] (DO[134])	UO[18] ACK8/SNO8	DO[18] (DO[138])
out 19	Xm+2	2	DO[15] (DO[135])	UO[19] SNACK	DO[19] (DO[139])
out 20	Xm+2	3	DO[16] (DO[136])	UO[20] Reserve	DO[20] (DO[140])
out 21	Xm+2	4	DO[17] (DO[137])	DO[1] (DO[121])	DO[21] (DO[141])
: out (8×A+B+1) :	Xm+A	B	DO[8×A+B-3] (DO[8×A+B+117])	DO[8×A+B-19] (DO[8×A+B+101])	DO[8×A+B+1] (DO[8×A+B+121])

*1 : in2 is also assigned to UI[4](CSTOPI).
*2 : in3 is also assigned to UI[17](PNSTROBE)

3.1.1 Digital I/O

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

Configuring I/O

In digital I/O, the configuration of the signal lines can be redefined. The following items are set. Refer to "3.1 I/O" for detail of configuring I/O.

> ⚠ **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 kind of I/O module.

SLOT

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

START

START assigns the logical number to the physical number to map the signal lines. The first physical number in the assignment should be specified.

> **NOTE**
> 1 A physical number specifies the pin of Input/Output lines on the I/O module. Logical number is assigned to this physical number. And each signal line can be assigned respectively.
> 2 Any physical number can be specified as the start point. Not assigned signal is automatically assigned to other logical number.

I/O configuration can be done with I/O configuration screen and I/O detail screen. When the assignment or settings of I/O is changed, cycle power of the controller to use new information. When the kind of I/O board is changed to the different one, I/O configuration may be done again.

Attribution of I/O signal

- Polarity

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

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

- Complementary

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

- Skipping simulated signals

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

Output

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

Simulated input/output

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

Procedure 3-1 Configuring Digital I/O

Step

1. Press the [MENU] key. The screen menu will be displayed.
2. Select “5 I/O”.
3. Press the F1, [TYPE]. The screen change menu will be displayed.
4. Select “Digital.”

Digital I/O list screen

```
I/O Digital Out
            #   SIM  STATUS                 1/512
       DO[   1]  U     OFF  [                 ]
       DO[   2]  U     OFF  [                 ]
       DO[   3]  U     OFF  [                 ]
       DO[   4]  U     OFF  [                 ]
       DO[   5]  U     OFF  [                 ]
       DO[   6]  U     OFF  [                 ]
       DO[   7]  U     OFF  [                 ]
       DO[   8]  U     OFF  [                 ]
       DO[   9]  U     OFF  [                 ]
       DO[  10]  U     OFF  [                 ]
       DO[  11]  U     OFF  [                 ]
  Sorted by port number.

   [ 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 list screen, press the F2, "MONITOR".

Digital I/O configuration screen

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







      Device Name : PrcI/O JB

   [ TYPE ]   LIST   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.
   d) When the entered values are valid, abbreviation PEND is displayed in STAT..
      If any entered value is invalid, abbreviation INVAL is displayed in STAT..

   Unnecessary lines can be deleted by pressing the F4 "DELETE".
   The abbreviations that will appear in Status mean the following:
   ACTIV : This assignment is now in use.
   PEND : Assignment is normal. Cycling power causes the ACTIV status to be entered.
   INVAL : A specified value is invalid.
   UNASG : No assignment has been made.
   PMC : Assignment has been made by PMC. This assignment cannot be changed in this screen.

8 To return to the list screen, press the F2, "LIST".

```
I/O Digital Out
            #   SIM  STATUS                 1/512
       DO[   1]  U     OFF  [DT SIGNAL 1      ]
       DO[   2]  U     OFF  [DT SIGNAL 2      ]
       DO[   3]  U     OFF  [DT SIGNAL 3      ]
       DO[   4]  U     OFF  [DT SIGNAL 4      ]

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

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

Digital I/O detail screen

```
I/O Digital Out
      Port Detail                                   1/3

         Digital Output                  [     1]

       1    Comment:  [                           ]

       2    Polarity:      NORMAL

       3  Complementary:    FALSE  [      1  -   2]

   [TYPE  | PRV-PT | NXT-PT |        |        |
```

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

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

11 To 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 the F3, "NXT-PT".

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

```
I/O Digital Out
             #   SIM  STATUS                   1/512
       DO[   1]  U      OFF  [DT SIGNAL 1      ]
       DO[   2]  U      OFF  [DT SIGNAL 2      ]
       DO[   3]  U      OFF  [DT SIGNAL 3      ]
       DO[   4]  U      OFF  [DT SIGNAL 4      ]

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

14 Cycle power of the controller so it can use the new information.

> ⚠ **WARNING**
> Cycling power is required to make a new setting valid. Otherwise, injury or property damage would occur.

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

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

```
I/O Digital Out
            #   SIM  STATUS                 1/512
       DO[   1]  S      ON  [DIGTAL    1      ]

   [ 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.

Configuring I/O

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. Refer to "3.1 I/O" for detail of configuring 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 kind of I/O module.

- **SLOT**

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

- **START PT**

  START PT assigns the logical number to the physical number to map the signal lines. The first physical number in the assignment should be specified.

> **NOTE**
> 1 A physical number specifies the Input/Output pin on the I/O module. Logical number is assigned to this physical number.
> 2 Any number can be specified as the start point of the physical number.
> 3 When two or more I/O boards are connected, the signal lines on the different boards can not be assigned 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 assigned 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, cycle power of 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 INSTRUCTIONS, and Section 6.4,MANUAL I/O CONTROL)

Execution of simulated I/O

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

Procedure 3-2 Configuring group I/O

Step

1. Press the [MENU] key. The screen menu will be displayed.
2. Select “5 I/O”.
3. Press the F1, [TYPE]. The screen change menu will be displayed.
4. Select "Group". Group I/O list screen will be displayed.

Group I/O list screen

```
I/O Group Out
            #   SIM   VALUE                   1/100
       GO[   1]  *      *   [                  ]
       GO[   2]  *      *   [                  ]
       GO[   3]  *      *   [                  ]
       GO[   4]  *      *   [                  ]
       GO[   5]  *      *   [                  ]
       GO[   6]  *      *   [                  ]
       GO[   7]  *      *   [                  ]
       GO[   8]  *      *   [                  ]
       GO[   9]  *      *   [                  ]
       GO[  10]  *      *   [                  ]
       GO[  11]  *      *   [                  ]
  Sorted by port number.

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

5. To switch the input screen to the output screen, or vice versa, press the F3, "IN/OUT".
6. To allocate I/O, press F2, "CONFIG".

Group I/O configuration screen

```
I/O Group Out
                                            1/100
      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
       10       0       0        0           0

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

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

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

> **NOTE**
> 1 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.
> 2 In the line of the group I/O which is assigned by PMC, “(PMC)” is displayed on the right of NUM PTS data. The configuration of the signal which is assigned by PMC cannot be changed.

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

Group I/O detail screen

```
I/O Group Out
      Port Detail                             1/1

         Group Output                 [    1]

       1    Comment:  [                          ]

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

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

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

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

11 When you are finished, press the [PREV] key to return to the list screen.

12 Cycle power off the controller so it can use the new information.

> ⚠ **WARNING**
> Cycling power is required to make a new setting valid. Otherwise, injury or property damage would occur.

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

3.1.3 Analog I/O

Analog I/O (AI/AO) signals are sent to and from the 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.

Configuring I/O

The physical numbers for the analog signal lines can be redefined. Refer to "3.1 I/O" for detail of configuring I/O.

> **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 I/O module.

- **SLOT**
  Indicates the number for the I/O module which compose RACK.

- **CHANNEL**
  Assigns 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 assigned to this physical number. This assignment 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, cycle power of 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 analog output can be set by executing the program or manual I/O control (Sections 4.6 and 6.4).

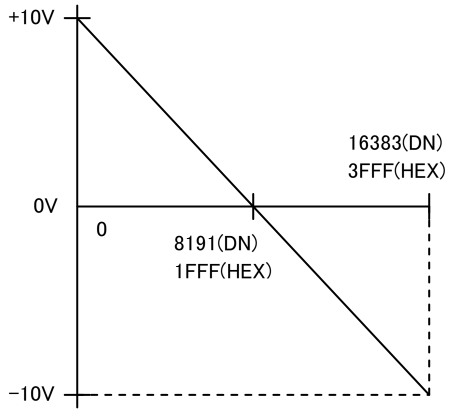
Execution of simulated I/O

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

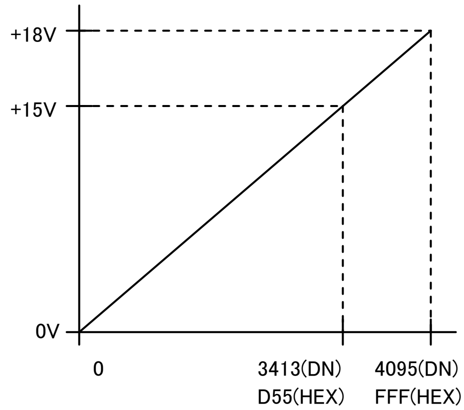
Conversion of analog I/O

For analog I/O of process I/O board, the digital value and voltage correspond as the following.

Process I/O analog Input

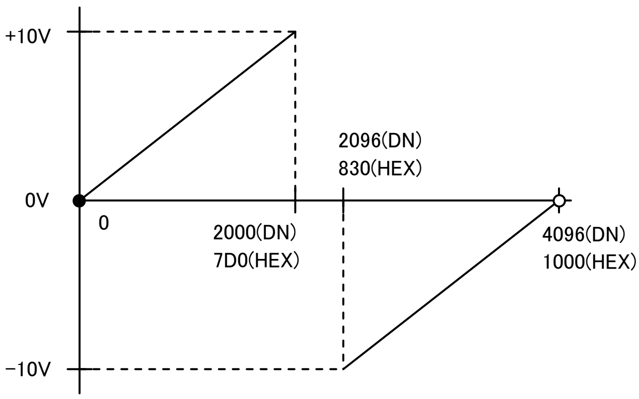


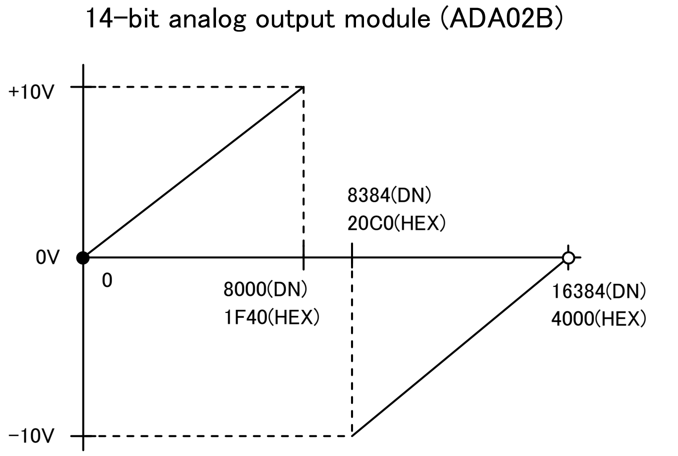
Process I/O analog output

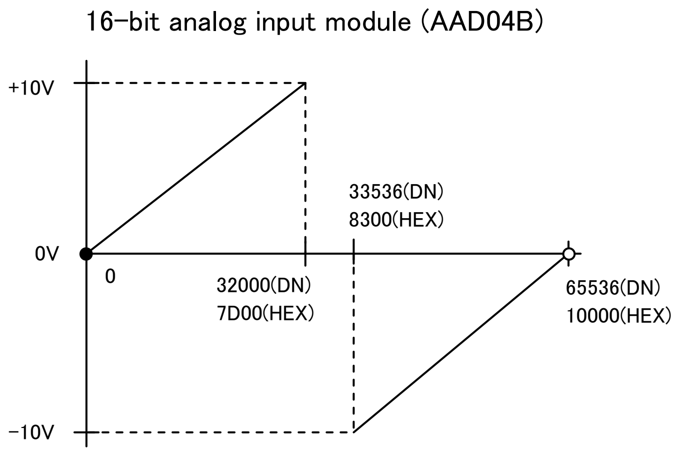


For analog I/O of I/O unit-MODEL A, the digital value and voltage correspond as the following.

12-bit analog input module (AAD04A)
12-bit analog output module (ADA02A)







If the setting is default, 16-bit module is unusable.
By following setting, 16-bit module is available.

$IOLNK[N].$RACK = (The rack number connecting to the module)
$IOLNK[N].$SLOT = (The slot number connecting to the module)
$IOLNK[N].$OPTION = 1
(N : arbitrary number)

Procedure 3-3 Configuring 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 [MENU] key. The screen menu will be displayed.
2. Select “5 I/O”.
3. Press the F1, [TYPE]. The screen change menu will be displayed.
4. Select "Analog". The analog I/O list screen will be displayed.

Analog I/O list screen

```
I/O Analog In
           #   SIM   VALUE                1/64
      AI[   1]  U      0   [                 ]
      AI[   2]  U      0   [                 ]
      AI[   3]  *      *   [                 ]
      AI[   4]  *      *   [                 ]
      AI[   5]  *      *   [                 ]
      AI[   6]  *      *   [                 ]
      AI[   7]  *      *   [                 ]
      AI[   8]  *      *   [                 ]
      AI[   9]  *      *   [                 ]
      AI[  10]  *      *   [                 ]
      AI[  11]  *      *   [                 ]
  Sorted by port number.

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

5 To switch the input screen to the output screen, press the F3, "IN/OUT".
6 To allocate I/O, press F2, "CONFIG".

Analog I/O configuration screen

```
I/O Analog In
                                         1/64
      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
      10     0       0         0

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

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

```
I/O Analog In
           #   SIM   VALUE                1/64
      AI[   1]  U      0  [analog sign1      ]
      AI[   2]  U      0  [analog sign2      ]
      AI[   3]  *      *  [analog sign3      ]
      AI[   4]  *      *  [analog sign4      ]

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

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

Analog I/O detail screen

```
I/O Analog In
      Port Detail                           1/1

         Analog Input                [    1]

       1    Comment:  [                          ]




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

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

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

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

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

13 Cycle power of the controller so that it can use the new information.

> ⚠ **WARNING**
> Cycling power is required to make a new setting valid. Otherwise, injury or property damage would occur.

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

3.2 ROBOT I/O

Robot I/O are digital signals which 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 operator's 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 Hand broken in Section 3.15, "SYSTEM CONFIG MENU".

- Abnormal air pressure input signal, *PPABN input

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

- *ROT input

The overtravel (robot overtravel) signal indicates an overtravel along each axis of the mechanical unit of the robot. In the normal status, the *ROT signal is on. When this signal is turned off, an alarm is generated and the robot is stopped immediately.
The *ROT input does not appear on the cable terminal of the end effector because it is processed within the mechanical unit of the robot.
While the *HBK or *ROT signal is off, the alarm state can temporarily be released by holding down [SHIFT] key and pressing the alarm release key. While holding down [SHIFT] key, move the tool to the appropriate position by jog feed.

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

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

Attribution of I/O signal

- Polarity

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

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

- Complementary

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

- Skipping simulated signals

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

Output

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

Simulated input/output

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

Procedure 3-4 Setting Robot I/O

Step

1 Press the [MENU] key. The screen menu will be displayed.
2 Select "5 I/O".
3 Press the F1 key, [TYPE]. The screen change menu will be displayed.
4 Select "Robot".

Robot I/O list screen

```
I/O Robot Out
            #   SIM  STATUS                1/8
       RO[   1]  U    OFF  [                  ]
       RO[   2]  U    OFF  [                  ]
       RO[   3]  U    OFF  [                  ]
       RO[   4]  U    ON   [                  ]
       RO[   5]  U    ON   [                  ]
       RO[   6]  U    OFF  [                  ]
       RO[   7]  U    ON   [                  ]
       RO[   8]  U    OFF  [                  ]



  Sorted by port number.

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

5 To switch the input screen to the output screen, press the F3, "IN/OUT".
6 To set the attribute of I/O, press the [NEXT] key and press the F4, "DETAIL" of the next page.

Robot I/O detail screen

```
I/O Robot Out
      Port Detail                           1/3

         Robot Dig. Output               [    1]

       1    Comment:  [                        ]

       2    Polarity:      NORMAL

       3  Complementary:    FALSE  [      1  -   2]



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

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 the [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 the [PREV] key to return to the list screen.

```
I/O Robot Out
            #   SIM  Status                1/8
       RO[   1]  U    OFF  [                  ]
       RO[   2]  U    OFF  [                  ]
       RO[   3]  U    OFF  [                  ]
       RO[   4]  U     ON  [                  ]

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

10 Cycle power of the controller so it can use the new information.

> ⚠ **WARNING**
> Cycling power is required to make a new setting valid. Otherwise, injury or property damage would occur.

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

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

```
I/O Robot Out

      RO[   1]  U   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.

Refer to "3.8 SETTING AUTOMATIC OPERATION" for the setting to control the robot from outside.

Configuring I/O

When all I/O assignment is deleted and cycling power of the robot controller, the connected I/O devices are recognized and proper I/O assignment is automatically performed. In case that the configuration of I/O is set automatically, the configuration of peripheral I/O is set depending on the setting of the item “UOP auto assignment” in the system configuration screen.
There are seven types of UOP auto assignment as shown below and each type has different assignment of peripheral I/O signals (UOP).

Table 3.3 UOP auto assignment

Type	UOP assignment type	I/O device to which UOP is assigned
None	No assignment	None
Full	Full assignment	I/O link master interface, etc
Full (Slave)	Full assignment	I/O link slave interface
Full (CRMA16)	Full assignment	R-30*i*B Mate main board (CRMA16)
Simple	Simple assignment	I/O link master interface, etc
Simple (Slave)	Simple assignment	I/O link slave interface
Simple (CRMA16)	Simple assignment	R-30*i*B Mate main board (CRMA16)

In default setting, the UOP auto assignment is set as follows.

- R-30*i*B : Full
- R-30*i*B Mate LR Tool : Simple(Slave)
- R-30*i*B Mate LR Handling Tool : Simple(CRMA16)

Refer to "3.1 I/O" for the standard I/O assignment according to the setting of UOP auto assignment.

UOP assignment types

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

Full assignment

All peripheral I/O signals can be used.
Eighteen input physical signals and twenty output physical signals are assigned to peripheral I/O signals.

Simple assignment

The peripheral I/O signals that the number of signals is small can be used.
Eight input physical signals and four output physical signals are assigned to peripheral I/O signals.

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

UI[1]	IMSTP	Always ON *4
UI[2]	HOLD	Operable
UI[3]	SFSPD	Always ON *4
UI[4]	CSTOPI	Allocated to the same signal as in RESET *1
UI[5]	RESET	Operable
UI[6]	START	Operable
UI[7]	HOME	No allocation
UI[8]	ENBL	Operable
UI[9]	RSR1/PNS1/STYLE1	Operable as PNS1 *3
UI[10]	RSR2/PNS2/STYLE2	Operable as PNS2 *3
UI[11]	RSR3/PNS3/STYLE3	Operable as PNS3 *3
UI[12]	RSR4/PNS4/STYLE4	Operable as PNS4 *3
UI[13]	RSR5/PNS5/STYLE5	No allocation
UI[14]	RSR6/PNS6/STYLE6	No allocation
UI[15]	RSR7/PNS7/STYLE7	No allocation
UI[16]	RSR8/PNS8/STYLE8	No allocation
UI[17]	PNSTROBE	Allocated to the same signal as in START *2
UI[18]	PROD_START	No allocation

UO[1]	CMDENBL	Operable
UO[2]	SYSRDY	No allocation
UO[3]	PROGRUN	No allocation
UO[4]	PAUSED	No allocation
UO[5]	HELD	No allocation
UO[6]	FAULT	Operable
UO[7]	ATPERCH	No allocation
UO[8]	TPENBL	No allocation
UO[9]	BATALM	Operable
UO[10]	BUSY	Operable
UO[11]	ACK1/SNO1	No allocation
UO[12]	ACK2/SNO2	No allocation
UO[13]	ACK3/SNO3	No allocation
UO[14]	ACK4/SNO4	No allocation
UO[15]	ACK5/SNO5	No allocation
UO[16]	ACK6/SNO6	No allocation
UO[17]	ACK7/SNO7	No allocation
UO[18]	ACK8/SNO8	No allocation
UO[19]	SNACK	No allocation
UO[20]	RESERVE	No allocation

*1 Since CSTOPI and RESET are allocated to the same signal, reset input can forcibly terminate the program if "CSTOPI for ABORT" is enabled.
*2 Since PNSTROBE and START are allocated to the same signal, the program is selected at the rising edge (OFF→ON) of the START signal and the program is started at the falling edge (ON→OFF) of the START signal.
*3 Only PNS can be used as the program selection method in simple allocation (that START and PNSTROBE are allocated to the same signal). Even if the "program selection method" other than PNS is selected on the Prog Select screen, PNS is automatically selected during power-on.
*4 These signals are assigned to the internal I/O device (rack 35, slot 1) in which the signal is always on.
*5 Since PROD_START is not allocated in simple allocation, when "START for CONTINUE only" item in System Config menu is TRUE, the program cannot be started by peripheral I/O. Set the "START for CONTINUE only" item FALSE in simple allocation.

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

The immediate stop signal turns servo power off by the software.
The *IMSTP input is on in the normal status. When this signal is turned off, the following processing is performed:

- An alarm is generated and the servo power is turned off.

- The robot operation is stopped immediately. Execution of the program is also stopped.

> ⚠ **WARNING**
> The *IMSTP signal is controlled by software. Please use external emergency stop for safety-critical processing. For connection of external emergency stop signal, refer to the "FANUC Robot series R-30*i*B/R-30*i*B Plus CONTROLLER MAINTENANCE MANUAL" (B-83195EN) or the "FANUC Robot series R-30*i*B Mate/R-30*i*B Mate Plus CONTROLLER MAINTENANCE MANUAL" (B-83525EN).

*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 feed rate 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 feed rate override is reduced to the value specified for $SCR.$SFRUNOVLIM. When jog feed is executed, the feed rate override is reduced to the value specified for $SCR.$SFJOGOVLIM. When *SFSPD is off, the feed rate override cannot exceed these values.

> ⚠ **WARNING**
> The *SFSPD signal controls deceleration and stop by software. To stop the robot immediately for safety purposes, use safety fence signal. For connection of safety fence signal, refer to the FANUC Robot series R-30*i*B/R-30*i*B Plus CONTROLLER MAINTENANCE MANUAL (B-83195EN) or the FANUC Robot series R-30*i*B Mate/R-30*i*B Mate Plus CONTROLLER MAINTENANCE MANUAL (B-83525EN).

> **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.

ENBL input signal, ENBL, UI [8]

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

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

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

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

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

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

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

STYLE1 to STYLE8 UI [9-16] (Enabled in the remote state.)

These are STYLE number select signals. When the start signal is received, the STYLE1 to STYLE8 inputs are read to select a program, then the selected program is executed. When another program is being executed or temporarily stopped, these signals are ignored. (→ Subsection 3.8.3, STYLE)

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. (→ Subsection 3.8.2, Program Number Selection (PNS))

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

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

- When FALSE is selected for "START for CONTINUE" only on the Config system setting screen, the program selected using the teach pendant is executed from the line to which the cursor is positioned. A temporarily stopped program is also continued. (Default)

- When TRUE is selected for "START for CONTINUE" only on the Config system setting screen, a temporarily stopped program is continued. When the program is not temporarily stopped, it cannot be started.

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

CMDENBL input UO [1]

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

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

SYSRDY output UO [2]

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

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

PROGRUN output UO [3]

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

PAUSED output UO [4]

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

HELD output UO [5]

HELD is output when the hold button is pressed. It is not output when the hold button is released.
Otherwise, when the HOLD signal UI[2] is OFF from some peripheral devices, then HELD is output.

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

TPENBL output UO [8]

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

BATALM output UO [9]

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

⚠ **CAUTION**

In case that the BATALM signal is specified to include BZAL/BLAL of Pulsecoder, BATALM is output when at least one Pulsecoder of all axis of all motion group detects BZAL/BLAL. However BATALM signal excludes Process axis (Servo torch axis) which has no motion group and Slave axis of Dual drive function.

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 ACK8 are used together with the function. When an RSR input is accepted, a pulse of the corresponding signal is output as an acknowledgment. The pulse width can be specified. (→ Subsection 3.8.1, Robot Service Request (RSR))

SNO1 to SNO8 outputs UO [11-18]

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

SNACK output UO [19]

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

Procedure 3-5 Assigning Peripheral I/O

Step

1 Press the [MENU] key. The screen menu will be displayed.
2 Select “5 I/O”.
3 Press the F1 key, [TYPE]. The screen change menu will be displayed.
4 Select "UOP".

Peripheral I/O list screen

```
I/O UOP In
            #         Status              1/18
       UI[   1]        OFF  [*IMSTP            ]
       UI[   2]        OFF  [*HOLD             ]
       UI[   3]        OFF  [*SFSPD            ]
       UI[   4]        OFF  [Cycle stop        ]
       UI[   5]        OFF  [Fault reset       ]
       UI[   6]        OFF  [Start             ]
       UI[   7]        OFF  [Home              ]
       UI[   8]        OFF  [Enable            ]
       UI[   9]        OFF  [RSR1/PNS1/STYLE1  ]
       UI[  10]        OFF  [RSR2/PNS2/STYLE2  ]
       UI[  11]        OFF  [RSR3/PNS3/STYLE3  ]
  Sorted by port number.

  [ TYPE ]  CONFIG   IN/OUT                    >
```

5 To switch the input screen to the output screen, or vice versa, press the F3, "IN/OUT".
6 To allocate I/O, press the F2, "CONFIG".

Peripheral I/O configuration screen

```
I/O UOP In
                                                 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





      Device Name : PrcI/O JB

   [ 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 STAT.
If any entered value is invalid, abbreviation INVAL is displayed in STAT.

Unnecessary lines can be deleted by pressing F4, "Delete".

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

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

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

Peripheral I/O detail screen

```
I/O UOP In
      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.

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 [PREV] key to return to the selection screen.
12 Cycle power of the controller so it can use the new information.

⚠ **WARNING**
Cycling power is required to make a new setting valid. Otherwise, injury or property damage would occur.

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

NOTE
To control the peripheral I/O (UOP) by integrated PMC (option), UOP is assigned to PMC address (F, G) by PMC internal I/O assignment. When UOP is assigned to PMC address, the mark "*" is displayed on the left of "UI" and "UO" in the peripheral I/O configuration menu. In this case, the setting of the peripheral I/O configuration menu is ignored, and the setting is not used for the assignment of UI and UO. The UI and UO are assigned to PMC address according to the setting of the PMC internal I/O assignment menu.

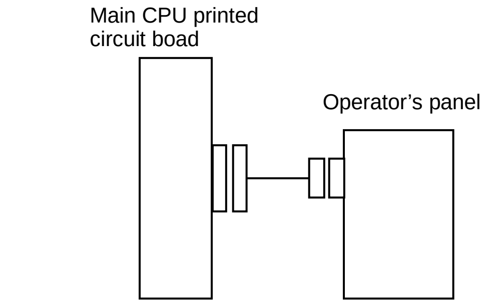
3.4 OPERATOR’S PANEL I/O

The operator’s panel I/O means dedicated digital signals for passing data indicating the status of the buttons and LEDs on the operator’s panel/box. The status of each input signal depends on whether the corresponding button on the operator’s panel is on or off. Each output signal is used to turn the corresponding LED lamp on the operator’s panel on or off.
For the operator’s panel I/O, the signal numbers cannot be mapped (redefined). Sixteen input and sixteen output signals are defined as standard. For the definition of the signals of the operator’s panel I/O, see Fig. 3.4.
When the operator’s panel is enabled, the operator’s panel I/O can be used to start a program. However, any signals which have a significant effect on safety are always enabled.
The operator’s panel is enabled when the following operator’s panel enable conditions are satisfied:

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

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

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



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

Fig. 3.4 Operator’s panel I/O

Table 3.4 (a) Operator’s panel input signals

Input signal	Description
***HOLD** SI [3] Always enabled. Not provided for the operator’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 [1] Always enabled.	The alarm release (fault reset) signal releases the alarm state. When the servo power is off, this signal turns on the servo power. In this case, the alarm state is not released until the servo power is turned on.
REMOTE SI [2] Always enabled. Not provided for the operator’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 "3.15, SYSTEM CONFIG MENU”.
START SI [6] Enabled in the operator’s panel enable state.	The start signal starts the currently selected program using the teach pendant from the line to which the cursor is positioned or restarts a temporarily stopped program. This signal functions at its falling edge when turned off after being turned on.

Table 3.4 (b) Operator's panel output signals

Output signal	Description
REMOTE SO [0] Not provided for the operator'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]	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 controller or the battery of the Pulsecoder of the robot. While keeping the power to the controller 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 [MENU] key to display the screen menu.
2 Select "5 I/O".
3 Press F1, [TYPE] to display the screen switching menu.
4 Select "SOP".

Operator's panel I/O list screen

```
I/O SOP Out
            #        STATUS                1/15
       SO[   0]          ON  [Remote LED      ]
       SO[   1]         OFF  [Cycle start     ]
       SO[   2]         OFF  [Hold            ]
       SO[   3]          ON  [Fault LED       ]
       SO[   4]          ON  [Batt alarm      ]
       SO[   5]         OFF  [                ]
       SO[   6]         OFF  [                ]
       SO[   7]          ON  [TP enabled      ]
       SO[   8]         OFF  [                ]
       SO[   9]         OFF  [                ]
       SO[  10]         OFF  [                ]
  Sorted by port number.

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

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

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

3.5 I/O LINK SCREEN

The I/O link screen can be used to make settings related to FANUC I/O Unit-MODEL B and display the configuration of the I/O link units.
The I/O link screen consists of the following screens:

- I/O link list screen
- Model B unit list screen
- 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 (master mode). 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 slot number.
The following figure is an example of the I/O link list screen when "process I/O board MA", one unit of "I/O Unit-MODEL B", and two units of "I/O Unit-MODEL A" are connected to the robot controller. The names of the I/O units are displayed in the order in which the units are connected to the robot controller.

```
I/O Link Device
                                              1/4
         Device Name    Comment          Rack  Slot
       1 PrcI/O MA   [                  ]  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 JA	Process I/O Board JA
PrcI/O JB	Process I/O Board JB
PrcI/O KA	Process I/O Board KA
PrcI/O KB	Process I/O Board KB
PrcI/O KC	Process I/O Board KC
PrcI/O NA	Process I/O Board NA
PrcI/O MA	Process I/O Board MA
PrcI/O MB	Process I/O Board MB
Model A	I/O Unit-MODEL A
Model B	I/O Unit-MODEL B
I/O adptr	I/O link connection unit
Other	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 unit list screen
I/O adptr	Signal count setting screen
Other	Signal count setting screen

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

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

```
I/O Link Device
  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 will be displayed. 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 will be displayed . This menu contains the following items. When no unit is set, ”*******” is displayed. ”*******” indicates that no unit is connected.

- BMD88P1
- BID16P1
- BOD16P1
- BIA16P1
- BMD88Q1

After a unit is set on this screen, the unit I/O can be used by cycling power.

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 [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 in the menu displayed by pressing [FCTN] key on this screen, 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" that cannot be used without setting the number of signals, set the number of signals on this screen.

When the cursor is positioned to ”I/O adptr” on the I/O link list screen, press the F3, "DETAIL" key. Then, "Signal count setting screen" will be displayed as shown below.

```
I/O Link Device
                                                1/2
  I/O adptr        Rack1   Slot1

       Port name                                Points
       1 Digital input                             0
       2 Digital Output                            0

    [ TYPE ]          LIST              CLR_ASG
```

Move the cursor to the number indicating the number of signals and enter a numeric value to set the number of signals.
The target I/O unit can be used by cycling power 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 in the menu displayed by pressing [FCTN] key on this screen, 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 will be displayed.

```
Clear ALL I/O assignments?
```

				YES	NO	

Press F4, "YES" to delete all assignment information. When cycling power, the assignment is set to the standard settings.

3.6 I/O CONNECTION FUNCTION

The I/O connection function enables the RI/DI/SI status to be output to DO/RO to report the signal input status to external devices.
The standard input/output ranges are shown below:

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

Explanation of the function/settings

Assign signals and enable or disable each assignment on Interconnect screen in I/O menu.
The following five types of screens are available:

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

DI DO connection setting screen (RI → DO)

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

DI DO connection setting screen (DI → RO)

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

DI DO connection setting screen (DI → DO)

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

DI DO connection setting screen (SI → DO)

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

DI DO connection setting screen (ES → DO)

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

List of ES signals

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

NOTE

Refer to the FANUC Robot series R-30*i*B/ R-30*i*B Plus CONTROLLER MAINTENANCE MANUAL (B-83195EN) or the FANUC Robot series R-30*i*B Mate/ R-30*i*B Mate Plus CONTROLLER MAINTENANCE MANUAL (B-83525EN) for more information on emergency stop signals.

Example)

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

NOTE

1. When DI[i] → DO[j] is set and this assignment is enabled, the status of DI[i] is output to DO[j] at regular intervals. Therefore, if the contents of DO[j] are changed using the teach pendant or a program, the change is not reflected.
2. Whether to enable or disable each assignment can be changed only on the setting screen, described above.
3. When different multiple input signals are assigned to the same output signal, the status of each input signal is output. For example, assume that the following settings are made:

   1 ENABLE RI[1] → DO[1]
   2 ENABLE RI[2] → DO[1]

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

Procedure 3-7 Setting the I/O connection function

Step

1 Press [MENU] key 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 will be displayed.

DI DO connection setting screen (RI→DO)

```
INTERCONNECT
                                            1/8
       No. Enb/Disabl   INPUT        OUTPUT
       1   DISABLE    RI[    1] ->  DO[    0]
       2   DISABLE    RI[    2] ->  DO[    0]
       3   DISABLE    RI[    3] ->  DO[    0]
       4   DISABLE    RI[    4] ->  DO[    0]
       5   DISABLE    RI[    5] ->  DO[    0]
       6   DISABLE    RI[    6] ->  DO[    0]
       7   DISABLE    RI[    7] ->  DO[    0]
       8   DISABLE    RI[    8] ->  DO[    0]

  [ TYPE ]          [SELECT]  ENABLE  DISABLE
```

5 Press F3, [SELECT].
6 Position the cursor to the screen to be displayed and press [ENTER] key or specify the item number of the screen to be displayed using a numeric key.

DI RO connection setting screen (DI→RO)

```
INTERCONNECT
                                            1/8
       No. Enb/Disabl   INPUT        OUTPUT
       1   DISABLE    DI[    0] ->  RO[    1]
       2   DISABLE    DI[    0] ->  RO[    2]
       3   DISABLE    DI[    0] ->  RO[    3]
       4   DISABLE    DI[    0] ->  RO[    4]
       5   DISABLE    DI[    0] ->  RO[    5]
       6   DISABLE    DI[    0] ->  RO[    6]
       7   DISABLE    DI[    0] ->  RO[    7]
       8   DISABLE    DI[    0] ->  RO[    8]

  [ TYPE ]          [SELECT]  ENABLE  DISABLE
```

3.7 SIMULATED INPUT SKIP FUNCTION

Overview

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

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

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

```
I/O Digital In
            #   SIM STATUS            1/512
       DI[   1] (S)   OFF  [                ]
       DI[   2]  U    ON   [                ]
       DI[   3]  U    ON   [                ]
       DI[   4]  U    OFF  [                ]
       DI[   5]  U    OFF  [                ]
       DI[   6]  U    OFF  [                ]
       DI[   7] (U)   OFF  [                ]
       DI[   8]  U    OFF  [                ]
       DI[   9]  U    OFF  [                ]
       DI[  10]  U    OFF  [                ]
       DI[  11]  U    OFF  [                ]
  Sorted by port number.

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

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

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

          [ OK ]
```

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

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

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

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

⚠ **WARNING**

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

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

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

Procedure for setting up the simulated input skip function

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

- Step

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

3.8 SETTING AUTOMATIC OPERATION

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

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

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

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

NOTE

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

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

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

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

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

NOTE

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

When the CMDENBL signal cannot be turned on, or the program cannot be started by peripheral I/O even if the CMDENBL signal is on, confirm the following items.

Item	Method to check
Check items when the CMDENBL cannot be turned on.	
- The remote conditions are satisfied.	SO[0:REMOTE] is on.
- The teach pendant enable switch is off.	SO[7:TPENBL] is off.
- The setting of Remote/Local is Remote.	SI[2:REMOTE] is on.
- The mode switch is set to AUTO mode.	SI[8] is on, and SI[9] is on.
- UI[3:SFSPD] is on.	UI[3:SFSPD] is on.
- UI[8:ENBL] is on.	UI[8:ENBL] is on.
- $RMT_MASTER = 0 (Peripheral equipment)	Check in the system variables screen.
- The operation enable conditions are satisfied.	
- Any alarm does not occur.	SO[3:FAULT] is off, UO[6:FAULT] is off.
- The servo power is turned on.	UO[2:SYSRDY] is on.
- The continuous operation mode is selected (the single step mode is disabled).	The STEP LED on the teach pendant is off. In case that the integrated PMC is enabled, INFO[1:STEP] is 0.
Check items when the program cannot be started even if the CMDENBL signal is on.	When the program is started by the peripheral I/O, if the alarm "SYST-011 Failed to run task" occurs, confirm the alarm detail code in alarm history screen.
- UI signals are enabled.	Check the setting of the system variables in the system variable screen.
- The PNS/RSR, PNSTROBE, PROD_START, START signals are input correctly.	Check the status of the signals in UI screen. Check the configuration of UI. In case that UI signals are displayed as "*UI", because the UOP signals are assigned to PMC, check the PMC program.
- The condition for automatic operation in the program select screen in setup menu are satisfied.	Check the setting in the program select screen. Example) If HOME position check is enabled, check whether the robot is in the home position.

Sequence of program restart by peripheral I/O

After eliminating the cause of an alarm, the program is restarted by peripheral I/O as follows.

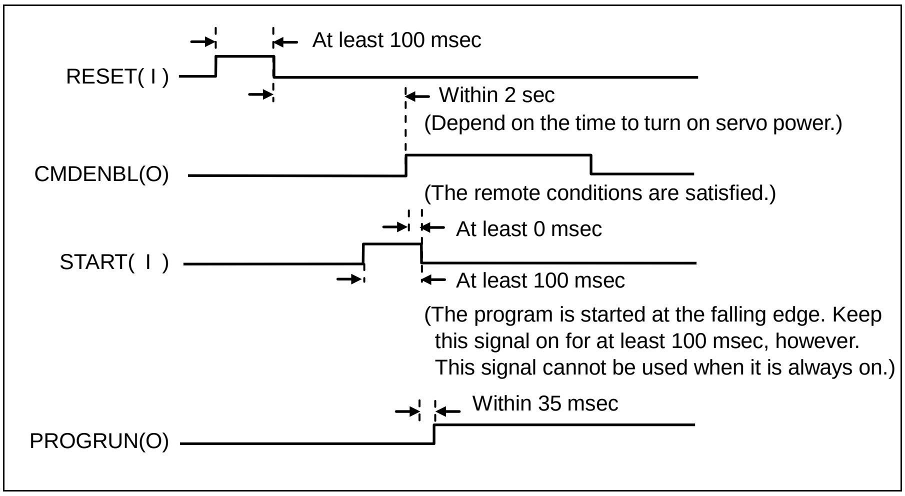


Fig. 3.8 Sequence of program restart by peripheral I/O

3.8.1 Robot Service Request (RSR)

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

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

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

2 Eight RSR registration numbers can be registered for RSR. The value obtained by adding a base number to an RSR registration number is used as the program number (four digits). For example, when RSR2 is input, the following value is used as the program number:
(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 RSR0121, not RSR121. If not, the robot will not operate.

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

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

4 When a program is in the terminated state, the selected program is started. When another program is being executed or is temporarily stopped, the request (job) is entered 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 the queue.

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

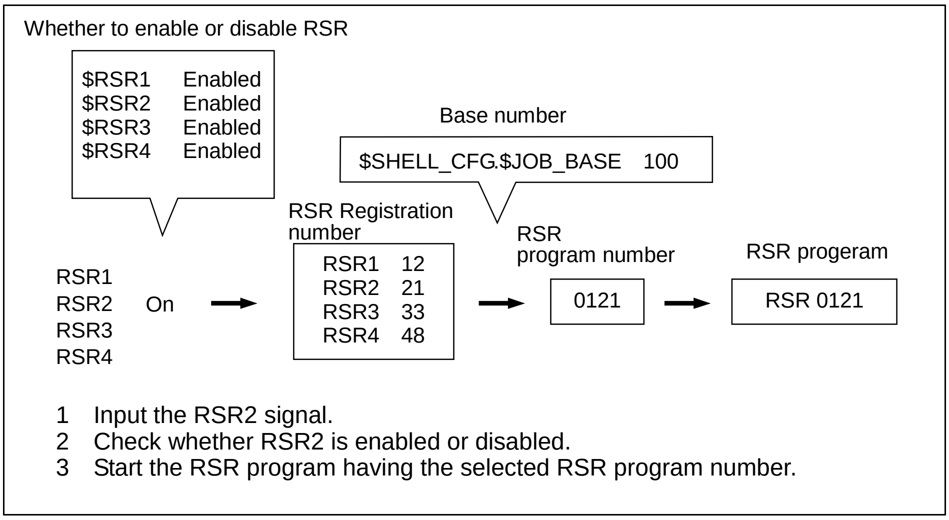


Fig. 3.8.1 (a) Robot service request

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

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

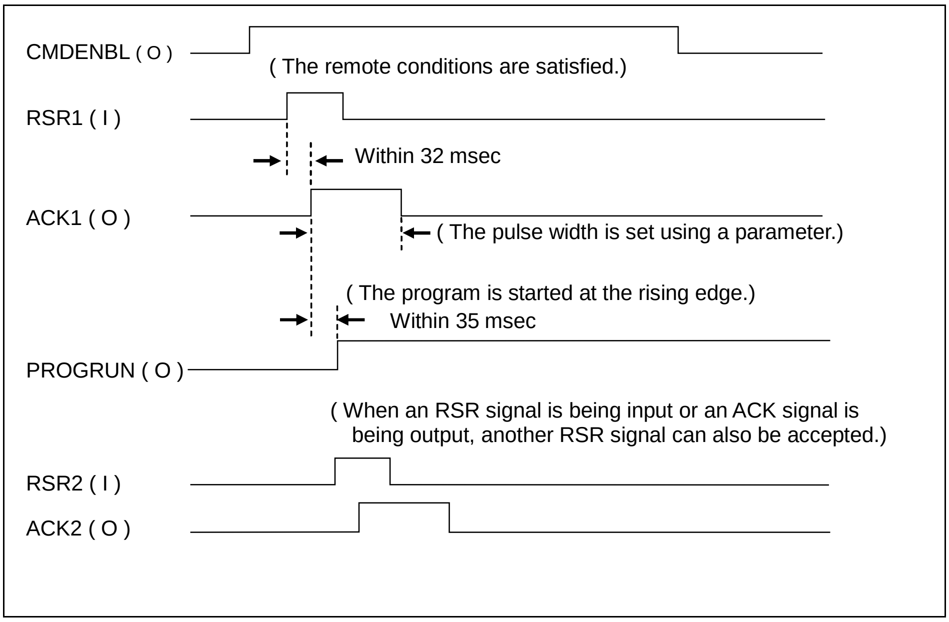


Fig. 3.8.1 (b) Sequence of automatic operation by RSR

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

Table 3.8.1 RSR setting items

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

Procedure 3-8 Setting RSR

Step

1. Press [MENU] key to display the screen menu.
2. Select “6 SETUP”.
3. Press F1, [TYPE] to display the screen switching menu.
4. Select Prog Select. The Prog Select screen appears.
5. Position the cursor to ”Program select mode”. Press F4, [CHOICE] and select "RSR", then press F3, "DETAIL".(Prog Select Screen →3.8.4)

RSR setting screen

```
Prog Select
                                                 1/12
         RSR Setup
       1 RSR1 program number  [ENABLE   ]  [    0]
       2 RSR2 program number  [ENABLE   ]  [    0]
       3 RSR3 program number  [ENABLE   ]  [    0]
       4 RSR4 program number  [ENABLE   ]  [    0]
       5 RSR5 program number  [ENABLE   ]  [    0]
       6 RSR6 program number  [ENABLE   ]  [    0]
       7 RSR7 program number  [ENABLE   ]  [    0]
       8 RSR8 program number  [ENABLE   ]  [    0]
       9 Job prefix                              [RSR]
      10 Base number                           [    0]
      11 Acknowledge function                [FALSE ]
      12 Acknowledge pulse width(msec)         [ 400]

   [ TYPE ]                    ENABLE   DISABLE
```

6. Position the cursor to the target item and enter a value.
7. After changing Program select mode, to enable the change, cycle power.

> ⚠ **WARNING**
> After the type of automatic operation function is changed, cycling power of the controller is required to enable the change. If not, the setting is not accepted.

3.8.2 Program Number Selection (PNS)

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

Step

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

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

2 The data of signals PNS1 to PNS8 is converted into a decimal PNS number. The sum of the PNS number and the reference number is a PNS program number (four digits).
(Program number)=(PNS number)+(Base number)
The specified PNS+(Program number) program number is named as follows.

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

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.

When a zero is input by the PNS1 to PNS8 inputs, the following process is done depending on the UOP allocation type.

UOP allocation type is "Full":

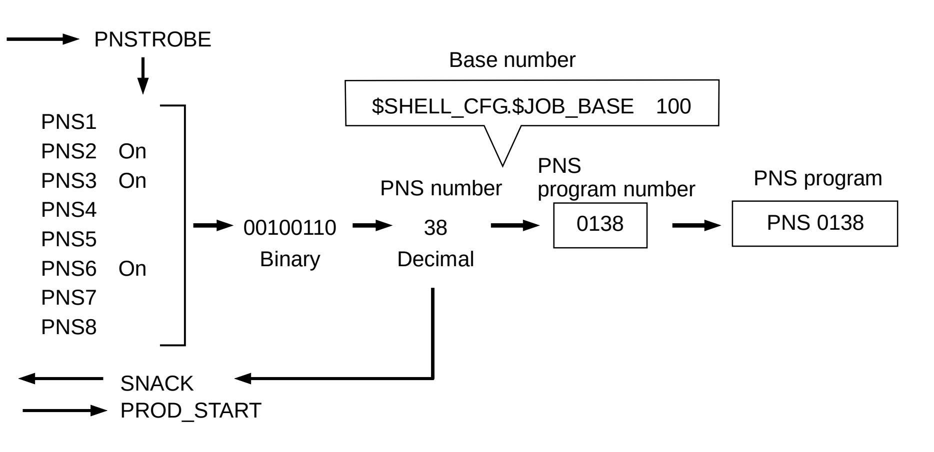
When a zero is input by the PNS1 to PNS8 inputs, the system enters the status in which no program is selected on the teach pendant.

UOP allocation type is "Simple":

When a zero is input by the PNS1 to PNS8 inputs, or the nonexistent program number is set and PNSTROBE signal is input, nothing is done.
When START is input in this state, if no program is selected, nothing is done. If a program is selected when START signal is input, the selected program is started.
(In case that the nonexistent program number is set and PNSTROBE signal is input, or START signal is input when no program is selected, the warning is displayed.)

3 SNO1 to SNO8 are output to indicate a PNS number as a binary code as confirmation. An SNACK pulse is output simultaneously. If the PNS number cannot be represented as an 8-bit numeric value, SNO1 to SNO8 output a zero.
4 The remote controller checks that the SNO1 to SNO8 output value is the same as the PNS1 to PNS8 input value when SNACK is output, and sends the automatic operation start input (PROD_START).
5 The controller receives the PROD_START input and starts the program.

Starting a program by PNS is enabled in the remote state.
Starting a program involving an operation (group) is enabled when the operation enable conditions as well as the remote conditions are satisfied.
The CMDENBL output is provided to indicate whether the above conditions are satisfied.



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.

Fig. 3.8.2 (a) Program number selection

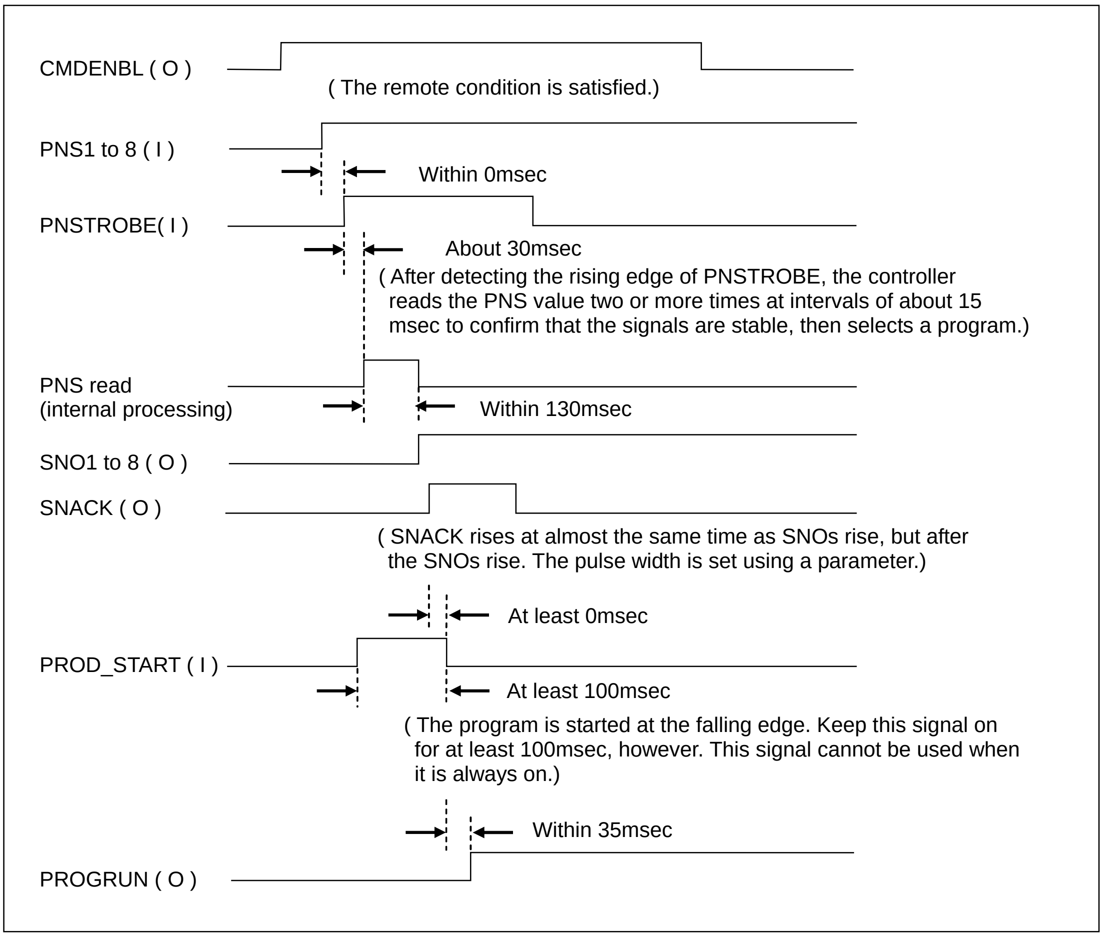


Fig. 3.8.2 (b) Sequence of automatic operation by PNS

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

Table 3.8.2 Setting the PNS function

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

Procedure 3-9 Setting the PNS function

Step

1. Press [MENU] key. The screen menu will be displayed.
2. Select “6 SETUP”.
3. Press the F1, [TYPE]. The screen change menu will be displayed.
4. Select Prog Select. Prog Select screen will be displayed.
5. Position the cursor to ”Program select mode”. Press F4, [CHOICE] and select "PNS", then press F3, "DETAIL".(Prog Select Screen →3.8.4)

PNS setting screen

```
Prog Select
                                              1/3
         PNS Setup
       1 Job prefix                         [PNS]
       2 Base number                      [  100]
       3 Acknowledge pulse width(msec)    [  200]

  [ TYPE ]
```

6. Place the cursor on a desired field and enter a value.
7. After changing RSR to PNS, to enable the change, cycle power of the controller.

> ⚠ **WARNING**
> After the type of automatic operation function is changed, cycling power is required to enable the change. If not, the setting is not accepted.

3.8.3 STYLE

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

Step

1. STYLE function needs programs to be set to each STYLE number in advance. Then program name is not restricted as PNS and RSR. (Refer to "Procedure 3-10 Setting" the style function about setting of the program.)
2. The controller reads the STYLE1 to STYLE8 input signals as a binary number. And the data of signals STYLE1 to STYLE8 is converted into a decimal STYLE number.
3. From the remote controller, the start input (START or PROD_START) is sent. Then program is selected by STYLE number, and selected program is started.
4. SNO1 to SNO8 are output to indicate a STYLE number as a binary code as confirmation. An SNACK pulse is output simultaneously.

5 If the program is pausing, and the start input (Only START. PROD_START is unusable.) is sent, no program selection is made and execution is resumed.

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

Starting a program by STYLE is enabled in the remote state.
Starting a program involving an operation (group) is enabled when the operation enable conditions as well as the remote conditions are satisfied.
The CMDENBL output is provided to indicate whether the above conditions are satisfied.

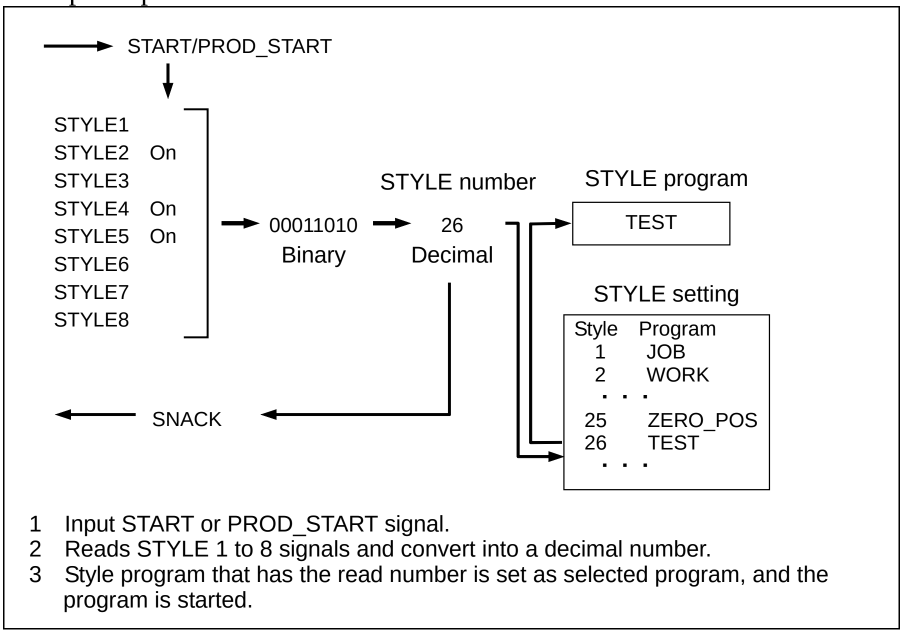


Fig. 3.8.3 (a) STYLE

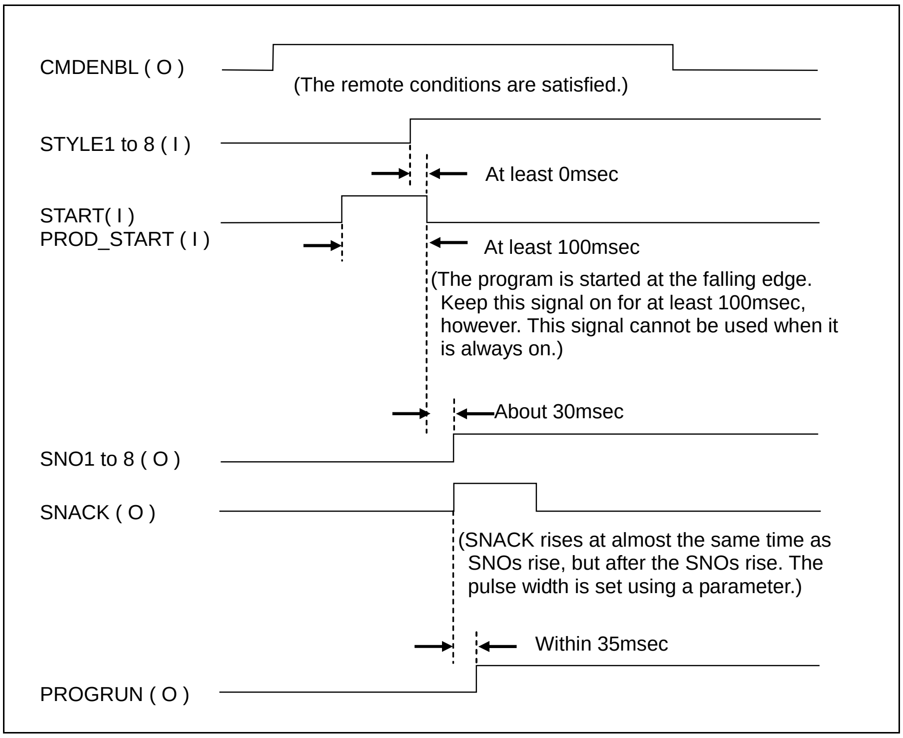


Fig. 3.8.3 (b) Sequence of automatic operation by STYLE

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

Table 3.8.3 Setting the STYLE function

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

Procedure 3-10 Setting the STYLE function

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select “6 SETUP”.
3 Press the F1, [TYPE]. The screen change menu will be displayed.
4 Select "Prog Select". Prog Select screen will be displayed.
5 Position the cursor to ”Program select mode”. Press F4, [CHOICE] and select "STYLE", then press F3, "DETAIL". (Prog Select Screen →3.8.4)

STYLE setting screen

```
Prog Select
     Style Table Setup                            1/32
     Style ProgName  Valid  Comment
       1    JOB        YES
       2    WORK       YES
       3    *******>   YES
       4    *******>   YES
       5    *******>   YES
       6    *******>   YES
       7    *******>   YES
       8    *******>   YES
       9    *******>   YES
      10    *******>   YES

   [ TYPE ]  CLEAR   DETAIL  [CHOICE]          >
```

```
Prog Select
      Style Config Setup                           1/3

       1 Acknowledge function:        [FALSE]
       2 Acknowledge pulse width(msec):[ 400]
       3 Max number in style table:     [ 32]

   [ TYPE ]                   TRUE    FALSE     >
```

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

8 After changing setting from RSR/PNS/OTHER to STYLE, to enable the change, cycle power of the controller.

> ⚠ **WARNING**
> After the type of automatic operation function is changed, cycling power of the controller to enable the change. If not, the setting is not accepted.

3.8.4 Prog Select Screen

Overview

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

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

Prog select screen

```
Prog Select
                                              1/13
      1 Program select mode:        STYLE
      2 Production start method:    UOP
        Production checks:
      3   At home check:            ENABLE
      4   Resume position tolerance: ENABLE
      5   Simulated I/O:            DISABLE
      6   General override < 100%:  DISABLE
      7   Prog override < 100%%:    DISABLE
      8   Machine lock:             DISABLE
      9   Single step:              DISABLE
     10   Process ready:            DISABLE
        General controls:
     11   Heartbeat timing:         1000  MS
     12   Low TEMP DRAM memory:     100 KB
     13   Low PERM CMOS memory:     50  KB

   [ TYPE ]            DETAIL  [CHOICE]   HELP
```

Program selection methods

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

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

Automatic operation start methods

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

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

* If the program selection method is either STYLE or OTHER and the automatic operation start method is UOP, it is possible to select and start a program with either UI[6:START] or UI[18:PROD_START], provided that a program is terminated. In this case, the program is started at the first line.
If a program is pausing, the program can be resumed with either UI[6:START] or UI[18:PROD_START]. In this case, no program selection is made.
Note that if the program selection method is either RSR or PNS, the effects of UI[6:START] and UI[18:PROD_START] are the same as those in the past.

List of program selection methods and automatic operation start methods

<table>
<tr><th colspan="2" rowspan="2"></th><th colspan="2">Automatic operation start method</th></tr>
<tr><th>UOP</th><th>Other</th></tr>
<tr><td rowspan="4">Program selection method</td><td>RSR</td><td>Program selection/start : RSR1 to 8</td><td>Not available.</td></tr>
<tr><td>PNS</td><td>Program selection : PNS1 to 8, PNSTROBE
Program start : UI[18:PROD_START]
or UI[6:START]</td><td>Not available.</td></tr>
<tr><td>Style</td><td>Program selection:STYLE1 to 8
Program start : UI[18:PROD_START]
or UI[6:START]</td><td>Program selection:STYLE1 to 8
Program start :$SHELL_WRK.$CUST_START</td></tr>
<tr><td>Other</td><td>Program selection: $SHELL_WRK.$CUST_NAME
Program start : UI[18:PROD_START]
or UI[6:START]</td><td>Program selection:
$SHELL_WRK.$CUST_NAME
Program start :$SHELL_WRK.$CUST_START</td></tr>
</table>

Automatic operation check

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

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

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

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

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

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

```
Prog Select DETAIL
                                                1/3
      Check:   At home check

       1 Check when run:              ENABLE
         Check when resume:           DISABLE
       2 Prompt if failure:           DISABLE
         Post error if failure:       ENABLE
       3 Post warning if forced:      DISABLE
         Force condition:             DISABLE

   [ TYPE ]                 ENABLE   DISABLE
```

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

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

- At home check

Condition	Action in the event of an error

Prompt if failure	The prompt below appears.
	`Robot arm is not at home.`
	`Jog or move robot to a home position.`
	`[OK]`

Force condition	
Post warning if forced	

- Resume position toler.

Condition	Action in the event of an error

Prompt if failure	The prompt below appears.
	`Robot arm moved too far.`
	`ANSWERING CONTINUE WILL`
	`INITIATE MOTION.`
	`CONTINUE [STOP]`
	When "CONTINUE" is selected, the program is executed with the warning "SYST-104 Resume tolerance ignored".

Force condition	
Post warning if forced	

- Simulated I/O

Condition	Action in the event of an error

Prompt if failure	The prompt below appears.
	`Simulated I/O ports exist`
	`(set from the I/O screens)`
	`CONTINUE FORCE [CANCEL]`
	• When "CONTINUE" is selected, the program continues to run.
	• If "FORCE" is selected, simulated I/O is forcibly canceled and the program is executed.
	• When "CANCEL" is selected, the program does not start but stops.

Force condition	Simulated I/O is automatically forcibly canceled and the program runs.
Post warning if forced	Simulated I/O is automatically forcibly canceled and the program runs, and the message below appears. "SYST-084 I/O forced unsimulated"

- General override < 100%

Condition	Action in the event of an error

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

Force condition	The program automatically runs at an override of 100%.

- Prog override < 100%

Condition	Action in the event of an error

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

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

- Machine lock

Condition	Action in the event of an error

Prompt if failure	The prompt below appears.
	`Motion is disabled`
	`CONTINUE [STOP]`
	• When "CONTINUE" is selected, the program starts with the warning below.
	"SYST-108 Machine lock ignored"
	• When "STOP" is selected, the program does not start but stops.

Force condition	
Post warning if forced	

- Single step

Condition	Action in the event of an error

Prompt if failure	The prompt below appears.
	`Single step is enabled (set from STEP TP key)`
	`CONTINUE RECHECK [STOP]`
	• When "CONTINUE" is selected, the program starts with the warning "SYST-109 Single step ignored".
	• When "RECHECK" is selected, the single step check is performed again, and in case of other than a single step, the program is started.
	• When "STOP" is selected, the program does not start but stops.

Force condition	A program start is made in the same way as when "CONTINUE" is selected and, at the same time, the single step is automatically canceled.
Post warning if forced	Single step is automatically canceled with the warning below, and the program start. "SYST-092 Single step forced off"

- **Process ready**

Condition	Action in the event of an error

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

Force condition	
Post warning if forced	

General settings

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

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

3.8.5 Cell Interface I/O

Cell interface I/O

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

Cell interface input signals

Cell interface input signals are explained in Table 3.8.5 (a).
To configure cell interface I/O, use Procedure 3-11.

Table 3.8.5 (a) Cell interface input signals

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

Cell interface output signals

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

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

Cell interface output signals are explained in Table 3.8.5 (b).
To configure cell interface I/O, use Procedure 3-11.

Table 3.8.5 (b) Cell interface output signals

Output signal	Explanation
Input Simulated	This output is used to notify PLC that there is a simulated input signal.
Output Simulated	This output is used to notify PLC that there is a simulated output signal.
OVERRIDE=100	This output is used to notify PLC that the override for the teach pendant is at 100%.
Robot ready	This output is used to notify PLC whether the CMDENBL, SYSRDY, and other conditions (whether each group operation is enabled, whether welding is enabled, and user-specified DI/O[] and RI/O[] conditions) are acceptable to a production start. The conditions contained in the signal must be previously set with "Status screen/robot ready". It is possible to check the status of each condition on this status screen.
Tryout Status	Used to notify PLC of the tryout mode status. ON = Tryout mode enabled
Heartbeat	This signal switches between ON and OFF after each heartbeat signal cycle. This signal is used to check the status of communication between the robot and PLC.
MH Fault (*)	If the controller is not in tryout mode, this signal turns ON if an alarm of the material handling function is generated. FAULT RESET causes this output to turn OFF.
MH Alert (*)	This output turns ON if a disabled alarm is generated within 20 cycles. FAULT RESET causes this output to turn OFF.
Refpos1[n]	If REF POS is enabled, this output turns ON if the robot is at reference position n of operation group 1.

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

Procedure 3-11 Configuring cell interface I/O

Step

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

```
I/O Cell Inputs
                                               1/1
           INPUT SIGNAL        TYPE # SIM  STATUS
       1  Tryout Mode         DI[   0]  U    ***

   [ TYPE ]  CONFIG   IN/OUT    SIM     UNSIM>   >
```

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

```
I/O Cell Outputs
                                               1/10
          Output signal         Type #  SIM STATUS
      1 Set if INPUT SIMULATED  DO[    0]U    ***
      2 Set if OUTPUT SIMULATED DO[    0]U    ***
      3 OVERRIDE = 100          DO[    0]U    ***
      4 In cycle                DO[    0]U    ***
      5 Abort Program           DO[    0]U    ***
      6 Tryout Status           DO[    0]U    ***
      7 Heartbeat signal        DO[    0]U    ***
      8 MH Fault                DO[    0]U    ***
      9 MH Alert                DO[    0]U    ***
     10 Robot motion G1         DO[    0]U    ***

 [ TYPE ]  CONFIG   IN/OUT    SIM     UNSIM    >
```

4 To specify whether to place an I/O signal in a simulated status, position the cursor on the SIM field for that I/O signal.
- To place the signal in a simulated status, press F4, "SIM". The signal is placed in a simulated status.
- To release a signal from a simulated status, press F5, "UNSIM". The signal is released from a simulated status.

5 To forcibly turn an I/O signal ON or OFF, position the cursor on the STATUS field for that I/O signal.
- To turn the I/O signal ON, press F4, "ON".
- To turn the I/O signal OFF, press F5, "OFF".

6 To allocate the signal, press F2, "CONFIG". The screen below appears.

```
I/O Cell Outputs
       Output Signal Details                  1/10

         Signal name:      Input Simulated
       1 Input type/no:    DO[   0]


 [ TYPE ] PREV_IO  NEXT-IO            VERIFY
```

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

- If the type of a signal can be changed and is required to be changed, move the cursor to the TYPE field for that signal, press F4, [CHOICE], select an I/O type, and press [ENTER] key.
- To change the number of a non-UOP I/O signal, place the cursor on “Number”, input a signal number, and press [ENTER] key.
- To check whether allocation is enabled, press F5, "VERIFY".
  → If the signal exists and allocation is enabled, the message "Port assignment is valid" appears.

→ If allocation is not enabled, the message "Port assignment is invalid" appears. Re-input is necessary.
→ VERIFY does not perform a double allocation check.

- To display the detailed information for the previous I/O signal, press F2, "PREV_IO".
- To display the detailed information for the next I/O signal, press F3, "NEXT-IO".

About the cell output signal, Robot ready

This function is an option. To use function it is necessary to order "Production Display Improvement" option (J887). The cell output signal, **Robot ready**, notifies PLC whether the robot is a production start ready status. The Robot ready output signal does not turn ON unless the various check items listed in Table 3.8.5 (c) result in OK. The check items are performed with Robot ready on the status screen. It is necessary to set monitor items in advance in accordance with the actual system.
The display and change procedure is explained in Procedure 3-12.

Table 3.8.5 (c) Check items with Robot ready

Check item	Explanation
CMENABLE	Indicates whether the CMENABLE-related items are OK or NG. NG appears if UI allocation is not correct.
SYSRDY	Indicates whether the SYSRDY-related items are OK or NG. NG appears if UI allocation is not correct.
General	Indicates whether the robot operation, welding enable/disable, pressurization enable/disable, and other items are OK or NG. To set the items and check whether they are enabled or disabled, press F2, "CONFIG" to display the display screen.
User	Indicates the status of the I/O signals necessary for a production start, such as DI/DO and RI/RO. NG appears if UI allocation is not correct. To set the items and check whether they are enabled or disabled, press F2, "CONFIG" to display the display screen.

Procedure 3-12 Displaying the status check screen

Step

1 Press MENU key and select "STATUS".
2 Press F1, [TYPE] and select "Robot ready".

```
STATUS Robot ready
     OUTPUT: Robot ready DO[0]              1/21

       1 CMDENBL TP disabled              NG
       2 CMDENBL SI[2]=ON                OK
       3 CMDENBL SFSPD=ON                 NG
       4 CMDENBL ENBL=ON                 OK
       5 CMDENBL $RMT_MASTER=0           OK
       6 CMDENBL SYSRDY=ON                NG
       7 CMDENBL No active alarms         NG
       8 CMDENBL Not in single step      OK
       9 SYSRDY ENBL=ON                   NG
      10 SYSRDY GRP1 Servo ready         OK

   [ TYPE ]  CONFIG                          >
```

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

Changing monitor items (CMDENBL and SYSRDY conditions not changeable)

- Step

1 Position the cursor to a desired item and press F2, "CONFIG".
2 To monitor the item, select "YES". Otherwise, select "NO".

3 To return to the list, press F2, "LIST".

Adding monitor signals

- **Step**

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

3.8.6 Custom I/O

Custom I/O allows you to display a customized I/O screen with up to 20 entries for any kind of I/O. It might or might not be displayed, depending on your configuration.

Table 3.8.6 shows the Custom I/O Setup items. Use Procedure 3-13 to set up Custom I/O.

Table 3.8.6 Custom I/O setup items

Custom I/O type	Custom I/O number	Description
DI, DO, RO, RI, GO, GI, UO, UI, AO, AI, SO, SI	Any valid I/O Index	Each kind of I/O can be configured in the Custom I/O screen. All comments, simulation status, and state of the I/O will mimic the System I/O screens. All simulations and states can be changed in the Custom I/O screen. I/O comments are inserted based on the currently defined I/O comments and cannot be changed from the Custom I/O menu. I/O comments must be changed in the appropriate I/O screen.

Procedure 3-13 Setting Custom I/O

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select "I/O".
3 Press F1, [TYPE]. The screen change menu will be displayed.
4 Select "Custom I/O". If the screen has not been configured, a blank screen that you can configure as desired will be displayed. The following screens are examples of custom I/O screens.

Custom I/O screen

```
CUSTOM I/O
                                               1/18
         DESCRIPTION          TYPE #    S STAT
 Prg running                  UO[     3]U  OFF
 FAULT                        UO[     6]U  OFF
 Fault reset                  UI[     5]U  OFF
 Style Req. /Echo             GO[     1]U  ON
 Style Select                 GI[     1]U  ON
 Style Ack                    DO[     1]U  OFF
 Lv Pounce(ok to)             DI[     1]U  OFF
 Fault Led                    SO[     3]U  OFF
 Hold                         SI[     3]U  OFF

  [ TYPE ]         CONFIG     ON      OFF    >
   INSERT   DELETE            ON      OFF    >
```

5 To delete an item on the Custom I/O screen, move the cursor to the line that you want to delete, then press NEXT key and press F2, "DELETE".
6 To insert an item on the Custom I/O screen , press NEXT key and press F1, "INSERT". This will display the configuration screen to configure the kind and index of I/O you want on your Custom I/O screen. The following screen is an example.

Custom I/O setting screen

```
CUSTOM I/O CONFIG
    I/O Signal Details                      1/1

    Signal name:
    I/O type/no: UI[   5]








   [ TYPE ]                  [CHOICE] VERIFY
```

7 To configure an item on the Custom I/O screen , move the cursor to the line that you want to configure, and press F3, "CONFIG", to display the configuration screen to configure the kind and index of I/O you want on your Custom I/O screen.

8 Press F5, "VERIFY". This will indicate whether the I/O is valid and will retrieve the I/O comment associated with the I/O point.

9 When you have completed the configuration, press PREV key. This will insert your I/O point below the cursor on the main page.

3.9 SETTING COORDINATE SYSTEMS

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

Joint coordinate system

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

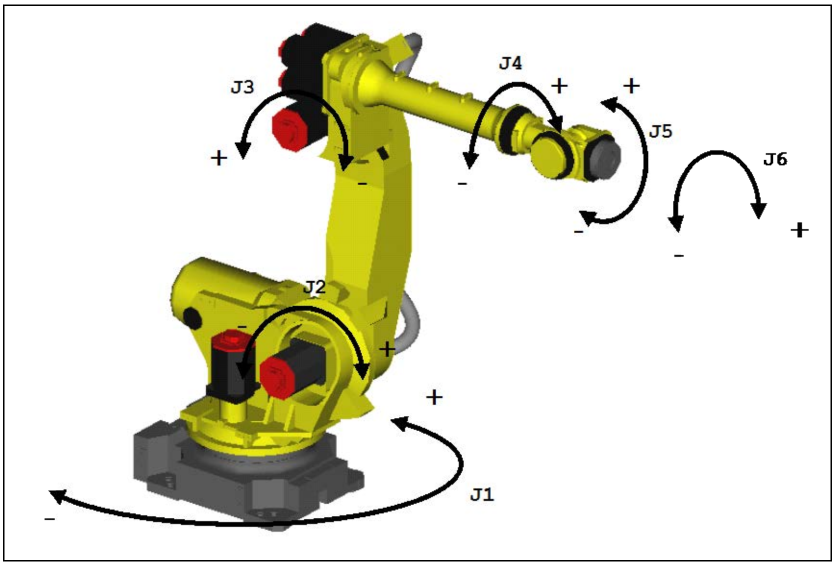


Fig. 3.9 (a) Joint coordinate system

Cartesian coordinate system

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

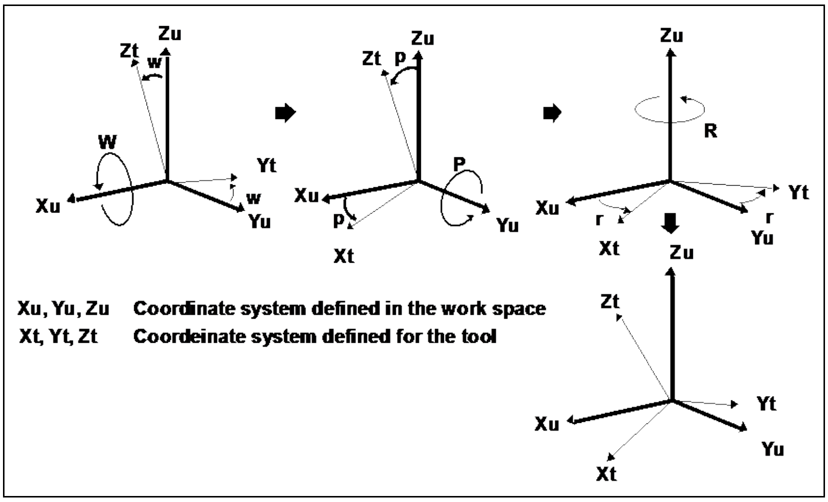


Fig. 3.9 (b) Meaning of (w, p, r)

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

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

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

Tool coordinate system

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

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

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

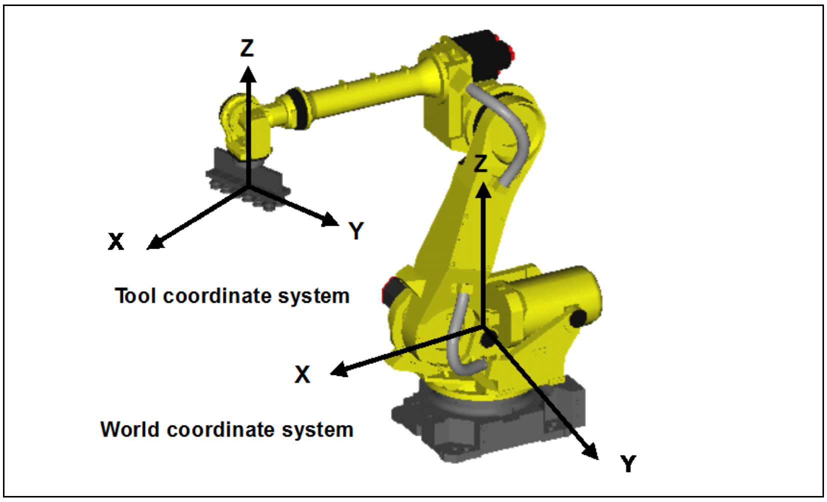


Fig. 3.9 (c) World and tool coordinate systems

User coordinate system

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

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

Jog coordinate system

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

Cell coordinate system

A coordinate system that all robots in the work cell share the original point. The cell coordinate system is used to represent the robot position in the work cell in 4D graphics function. By setting the cell coordinate system, the relation of the position between the robots can be represented. The cell coordinate system is defined by the position of the original point of the world frame in the cell coordinate system (x, y, z), and the rotation angle around X axis, Y axis and Z axis (w, p, r). The cell coordinate system is set in each group of all robots in the work cell.

Cell floor

A coordinate system to describe the floor in which the robot is placed in 4D graphics function. To set the cell floor, set the position and the attitude of the floor in the cell coordinate system. In default setting, the cell floor is set automatically depending on the robot model. The cell floor can be set in the coordinate system setting screen.

3.9.1 Setting a Tool Coordinate System

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

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

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

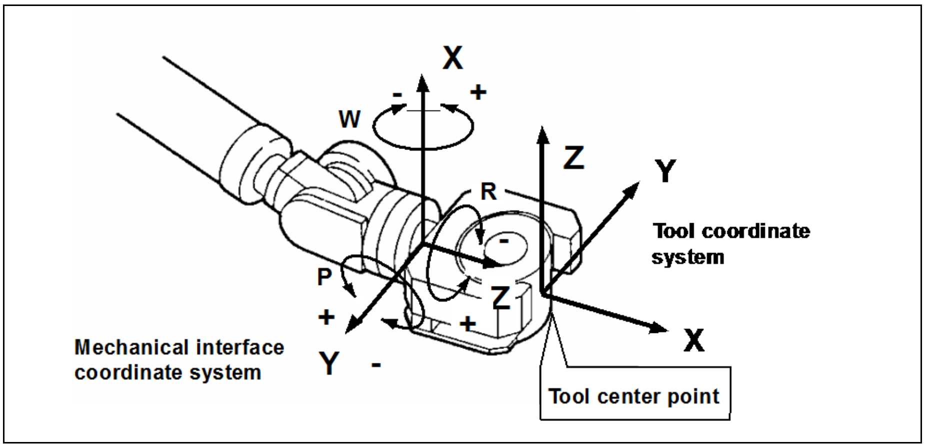


Fig. 3.9.1 (a) Tool coordinate system

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

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

> **NOTE**
> Depending on the combination of robot model and option, the tool coordinate system may be set in advance.

The number of the tool coordinate systems can be increased up to 253 by the following way. (up to 29 in R-30*i*B and R-30*i*B Mate.)

1. Perform a controlled start.
2. Press MENU key.
3. Select “4. Variables”.
4. Set the number of tool coordinate systems to the system variable $SCR.$MAXNUMTOOL. The maximum number of tool coordinate systems is 253. (The maximum number of tool coordinate systems is 29 in R-30*i*B and R-30*i*B Mate.)
5. Perform a cold start.

> **NOTE**
> If you set maximum number of tool coordinate system over 29 in R-30*i*B Plus, TP programs saved in the system cannot be load to either R-30*i*B or R-30*i*B Mate.

The tool frame can be set by five following methods.

Three Point Method (TCP auto set)

Use the three point method to define the tool center point (TCP).The three approach points must be taught with the tool touching a common point from three different approach statuses.
As a result, the location of TCP is automatically calculated.
To set the TCP accurately, three approach directions had better differ from others as much as possible.
In the three point method, only the tool center point (x, y, z) can be set. The setting value of the tool orientation (w, p, r) is the standard value (0, 0, 0). The tool orientation should be defined by the six point method or direct list method after the location is set.

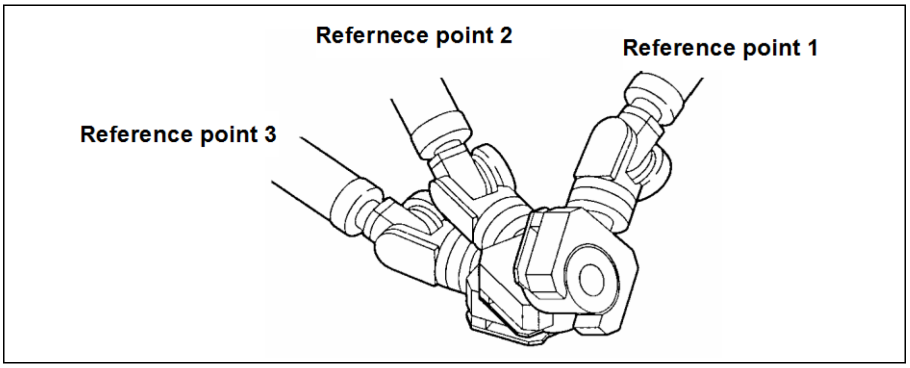


Fig. 3.9.1 (b) TCP auto set by the three point method

Six Point Method

The tool center point can be set in the same method as the three point method.
Then, set the tool attitude (w, p, r).

You can choose Six Point (XY) Method and Six Point (XZ) Method.
In Six Point (XZ) Method, for example, teach the robot so that w, p, and r indicate a given point in space, a point in the positive direction of the X-axis parallel to the tool coordinate system, and a point on the XZ plane. Also, teach the robot using Cartesian or tool jog so that the tilt of the tool does not change.

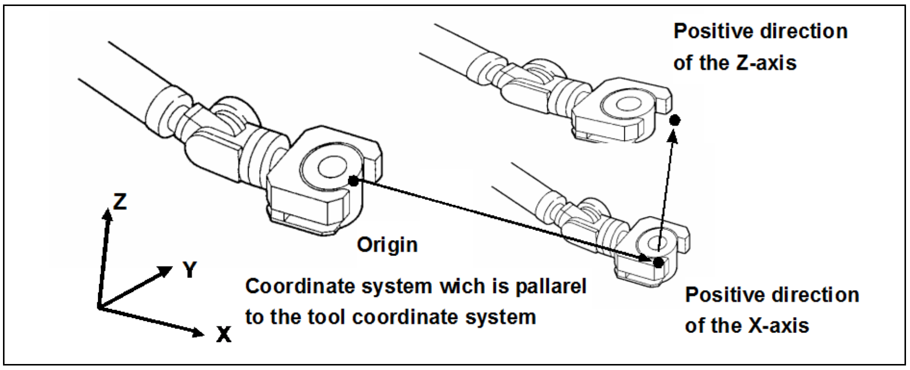


Fig. 3.9.1 (c) Six point (XZ) method

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.

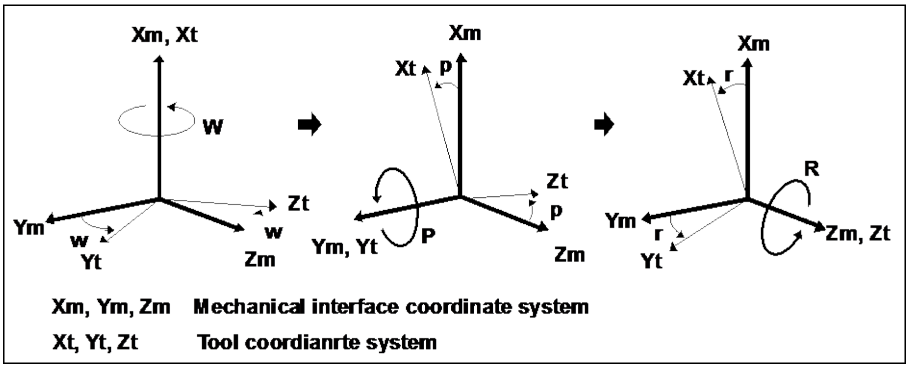


Fig. 3.9.1 (d) Meaning of (w, p, r) used in direct teaching method

Two Point + Z Method (Available on software version 7DC1/04 or later)

This function can set TCP for robots which cannot tilt their tools to the XY plane in the world frame (i.e. 4 axes robots). Teach the approach point 1 and 2 with the tool touching a certain fixed point from two different approach statuses. X and Y value are calculated from this two approach points. Measure Z value with a ruler, etc. and input Z value directly. Also, input w, p, and r value directly. (However, input 0 to all three values when the direction of the flange and the direction of the tool attitude are same.)

Four Point Method (Available on software version 7DC3/21 or later)

Use the four point method to set the tool center point (TCP) based on an error feedback. The four approach points must be taught with the tool touching a common point from four different approach statuses. As a result, not only the location (x, y, z) of the tool frame but also the deviation of positions of the tool tip at each approach point are automatically calculated. Users can improve inaccuracy of TCP, attributed to inaccurate teaching, by iteratively correcting indicated outliers. The setting value of the tool orientation (w, p, r) is the standard value (0, 0, 0).

Procedure 3-14 TCP auto set (Three Point Method)

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select “6 SETUP”.
3 Press the F1, [TYPE]. The screen change menu will be displayed.
4 Select "Frames".
5 Press F3, [OTHER] and then select "Tool Frame". Tool frame list screen will be displayed.

Tool frame list screen

```
SETUP Frames
      Tool Frame     / Direct Entry         1/10
            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      [            ]
      10 0.0      0.0      0.0      [            ]
     Active TOOL $MNUTOOLNUM[G:1] = 1

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

7 Move the cursor to the line of the tool frame number you want to set.
8 Press F2, "DETAIL". The tool frame setup screen of the selected frame number will be displayed.
9 Press F2, "METHOD" and then select "Three Point".

Tool frame setup screen (Three Point Method)

```
SETUP Frames
      Tool Frame         Three Point          1/4
      Frame Number:        1
        X:    0.0   Y:    0.0   Z:    0.0
        W:    0.0   P:    0.0   R:    0.0
       Comment:                TOOL1

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


     Active TOOL $MNUTOOLNUM[G:1] = 1

   [ TYPE ][METHOD]  FRAME
```

11 To add a comment:
  a Move the cursor to the comment line and press [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.

12 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 [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.

```
SETUP Frames
      Tool Frame         Three Point          3/4
      Frame Number:        1
        X:     0.0   Y:     0.0   Z:     0.0
        W:     0.0   P:     0.0   R:     0.0
       Comment:                TOOL1

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


     Active TOOL $MNUTOOLNUM[G:1] = 1

   [ 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
      Tool Frame         Three Point          4/4
      Frame Number:        1
        X:     0.0   Y:     0.0   Z:     0.0
        W:     0.0   P:     0.0   R:     0.0
       Comment:                TOOL1

       Approach point 1:   USED
       Approach point 2:   USED
       Approach point 3:   USED


     Active TOOL $MNUTOOLNUM[G:1] = 1

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

13 To move the robot to a recorded position, press and hold [SHIFT] key and press F4, "MOVE_TO".
14 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 [PREV] key.
15 To display the tool frame list screen, press [PREV] key. You can see the settings (x, y, z, and comment) for all tool frames.

```
SETUP Frames
      Tool Frame       / Three Point          1/10
           X        Y         Z       Comment
       1 100.0    0.0      120.0    [TOOL1       ]
       2 0.0      0.0      0.0      [            ]
       3 0.0      0.0      0.0      [            ]
       4 0.0      0.0      0.0      [            ]
       5 0.0      0.0      0.0      [            ]
       6 0.0      0.0      0.0      [            ]
       7 0.0      0.0      0.0      [            ]
       8 0.0      0.0      0.0      [            ]
       9 0.0      0.0      0.0      [            ]
      10 0.0      0.0      0.0      [            ]
     Active TOOL $MNUTOOLNUM [G:1] = 1

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

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

⚠ CAUTION

1 If you do not press F5, "SETIND", the tool frame will not be effective.
2 After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

NOTE

To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 ”Moving the Robot by Jog Feed”.

17 To delete the data of the set frame, move the cursor to the desired frame and press F4, "CLEAR". You can select whether to clear its comment too or not.

Procedure 3-15 Setting Up Tool Frame Using the Six Point (XZ) Method

The procedure of Six Point (XZ) Method is explained here. In Six Point (XY) Method, “Z Direction Point” is displayed as “Y Direction Point”, but the procedure is the same.

Step

1 Display the tool frame list screen (Refer to the three point method).

```
SETUP Frames
      Tool Frame       / Three Point          1/10
           X        Y         Z       Comment
       1 100.0    0.0      120.0    [TOOL1       ]
       2 0.0      0.0      0.0      [            ]
       3 0.0      0.0      0.0      [            ]
       4 0.0      0.0      0.0      [            ]
       5 0.0      0.0      0.0      [            ]
       6 0.0      0.0      0.0      [            ]
       7 0.0      0.0      0.0      [            ]
       8 0.0      0.0      0.0      [            ]
       9 0.0      0.0      0.0      [            ]
      10 0.0      0.0      0.0      [            ]
     Active TOOL $MNUTOOLNUM [G: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 will be displayed.
4 Press F2, "METHOD".

5 Select "Six Point (XZ)". The tool frame setup / six point (XZ) screen is displayed.

Tool frame setup screen (Six Point (XZ) Method)

```
SETUP Frames
      Tool Frame         Six Point(XZ)       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 $MNUTOOLNUMN[G: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 [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.

```
SETUP Frames
      Tool Frame         Six Point(XZ)       5/7
      Frame Number:         2
        X:    0.0   Y:    0.0   Z:    0.0
        W:    0.0   P:    0.0   R:    0.0
       Comment:   TOOL2
       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 $MNUTOOLNUMN[G: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
      Tool Frame         Six Point(XZ)       7/7
      Frame Number:         2
        X:    200.0   Y:    0.0   Z:    255.5
        W:    90.0   P:    0.0   R:    180.0
       Comment:   TOOL2
       Approach point 1:        USED
       Approach point 2:        USED
       Approach point 3:        USED
       Orient Origin Point:     USED
       X Direction Point:       USED
       Z Direction Point:       USED

     Active TOOL $MNUTOOLNUMN[G:1] = 1

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

7 Press [PREV] key. The tool frame list screen will be displayed. You can see all the tool frame settings.

```
SETUP Frames
      Tool Frame       / Six Point(XZ)       2/10
            X       Y         Z       Comment
       1 100.0    0.0      120.0    [TOOL1       ]
       2 200.0    0.0      255.5    [TOOL2       ]
       3 0.0      0.0      0.0      [            ]
       4 0.0      0.0      0.0      [            ]
       5 0.0      0.0      0.0      [            ]
       6 0.0      0.0      0.0      [            ]
       7 0.0      0.0      0.0      [            ]
       8 0.0      0.0      0.0      [            ]
       9 0.0      0.0      0.0      [            ]
      10 0.0      0.0      0.0      [            ]
     Active TOOL $MNUTOOLNUM [G:1] = 1

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

8 To make the set tool frame effective, press F5, "SETIND", then enter the frame number.

⚠ CAUTION

1 If you do not press F5, "SETIND", the tool frame will not be effective.
2 After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

NOTE

To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 "Moving the Robot by Jog Feed".

9 To delete the data of the set frame, move the cursor to the desired frame and press F4, "CLEAR". You can select whether to clear its comment too or not.

Procedure 3-16 Setting Up Tool Frame Using the Direct List Method

Step

Display the tool frame list screen (Refer to the three point method).

```
SETUP Frames
      Tool Frame       / Direct Entry        2/10
            X       Y         Z       Comment
       1 100.0    0.0      120.0    [TOOL1       ]
       2 200.0    0.0      255.5    [TOOL2       ]
       3 0.0      0.0      0.0      [            ]
       4 0.0      0.0      0.0      [            ]
       5 0.0      0.0      0.0      [            ]
       6 0.0      0.0      0.0      [            ]
       7 0.0      0.0      0.0      [            ]
       8 0.0      0.0      0.0      [            ]
       9 0.0      0.0      0.0      [            ]
      10 0.0      0.0      0.0      [            ]
     Active TOOL $MNUTOOLNUM [G: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 [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 will be displayed.

Tool frame setup screen (Direct List Method)

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

     Active TOOL $MNUTOOLNUM[G:1] = 1

   [ TYPE ][METHOD]  FRAME
```

6 Add a comment. Refer to TCP auto set (Three Point Method) for details.
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 [ENTER] key. A new numerical value is set.

```
SETUP Frames
      Tool Frame          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
          Configuration:      N D B, 0, 0, 0

     Active TOOL $MNUTOOLNUM[G:1] = 1

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

8 To display the tool frame list screen, press [PREV] key. You can see the settings of all the tool frame.

```
SETUP Frames
      Tool Frame        / Direct Entry        3/10
            X        Y        Z       Comment
       1 100.0    0.0      120.0    [TOOL1        ]
       2 200.0    0.0      255.5    [TOOL2        ]
       3 0.0      0.0      350.0    [TOOL3        ]
       4 0.0      0.0      0.0      [             ]
       5 0.0      0.0      0.0      [             ]
       6 0.0      0.0      0.0      [             ]
       7 0.0      0.0      0.0      [             ]
       8 0.0      0.0      0.0      [             ]
       9 0.0      0.0      0.0      [             ]
      10 0.0      0.0      0.0      [             ]
     Active TOOL $MNUTOOLNUM [G:1] = 1

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

9 To make the set tool frame effective, press F5, "SETIND", then enter the frame number.

⚠ CAUTION

1 If you do not press F5, "SETIND", the tool frame will not be effective.
2 After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

NOTE

To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 ”Moving the Robot by Jog Feed”.

10 To delete the data of the set frame, move the cursor to the desired frame and press F4, "CLEAR". You can select whether to clear its comment too or not.

Procedure 3-17 Setting Up Tool Frame Using the Two Point + Z

Step

1 Display the tool frame list screen (Refer to the three point method).

```
SETUP Frames
      Tool Frame       / Direct Entry        4/10
            X       Y        Z       Comment
       1 100.0    0.0      120.0    [TOOL1        ]
       2 200.0    0.0      255.5    [TOOL2        ]
       3 0.0      0.0      350.0    [TOOL3        ]
       4 0.0      0.0      0.0      [             ]
       5 0.0      0.0      0.0      [             ]
       6 0.0      0.0      0.0      [             ]
       7 0.0      0.0      0.0      [             ]
       8 0.0      0.0      0.0      [             ]
       9 0.0      0.0      0.0      [             ]
      10 0.0      0.0      0.0      [             ]
     Active TOOL $MNUTOOLNUM [G: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 "Two Point + Z". The tool frame setup is displayed, and Z, W, P, and R contain present tool frame value.

Tool frame setup screen (Two Point + Z)

```
SETUP Frames
      Tool Frame          Two Point + Z        1/7
      Frame Number:        4
        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
       Z:                      0.000
       W:                      0.000
       P:                      0.000
       R:                      0.000

     Active TOOL $MNUTOOLNUMN[G:1] = 1

   [ TYPE ][METHOD]  FRAME
```

6 Teach the 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 [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.

```
SETUP Frames
      Tool Frame          Two Point + Z        2/7
      Frame Number:        4
        X:    0.0   Y:    0.0   Z:    0.0
        W:    0.0   P:    0.0   R:    0.0
       Comment:                 TOOL4
       Approach point 1:     RECORDED
       Approach point 2:       UNINIT
       Z:                        0.000
       W:                        0.000
       P:                        0.000
       R:                        0.000

     Active TOOL $MNUTOOLNUMN[G:1] = 1

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

NOTE

- Set flanges upward or downward in case of teaching approach points. And, match the orientation of flanges of two approach points.
- Teach the approach point 1 and 2 at different locations.
- If input data is not applied to upper conditions, message “Invalid set of input points” is displayed on the coordinates screen.

7 After you input all approach points, approach points are changed to "USED" and the tool frame is calculated.

8 Measure Z value with a ruler, etc. and input Z value directly. Also, input W, P, and R value directly.

NOTE

When Z, W, P, and R are input, X and Y are changed if all approach points are RECORDED.

```
SETUP Frames
      Tool Frame          Two Point + Z        3/7
      Frame Number:        4
        X:   -5.5   Y:    6.6   Z:   10.0
        W:    0.0   P:    0.0   R:    0.0
       Comment:                 TOOL4
       Approach point 1:         USED
       Approach point 2:         USED
       Z:                       10.000
       W:                        0.000
       P:                        0.000
       R:                        0.000

     Active TOOL $MNUTOOLNUMN[G:1] = 1

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

9 To display the tool frame list screen, press [PREV] key. You can see the settings of all the tool frames.

```
SETUP Frames
      Tool Frame        / Two Point + Z        4/10
           X        Y         Z       Comment
       1 100.0    0.0      120.0    [TOOL1       ]
       2 200.0    0.0      255.5    [TOOL2       ]
       3 0.0      0.0      350.0    [TOOL3       ]
       4 -5.5     6.6      10.0     [TOOL4       ]
       5 0.0      0.0      0.0      [            ]
       6 0.0      0.0      0.0      [            ]
       7 0.0      0.0      0.0      [            ]
       8 0.0      0.0      0.0      [            ]
       9 0.0      0.0      0.0      [            ]
      10 0.0      0.0      0.0      [            ]
     Active TOOL $MNUTOOLNUM [G:1] = 1

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

10 To make the set tool frame effective, press F5, "SETIND", then enter the frame number.

⚠ CAUTION

1. If you do not press F5, "SETIND", the tool frame will not be effective.
2. After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

NOTE

To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 "Moving the Robot by Jog Feed".

11 To delete the data of the set frame, move the cursor to the desired frame and press F4, CLEAR. You can select whether to clear its comment too or not.

Procedure 3-18 Setting Up Tool Frame Using the Four Point Method

Step

1 Display the tool frame list screen (Refer to the three point method).

```
SETUP Frames
      Tool Frame     / Direct Entry            5/10
           X        Y         Z       Comment
       1 100.0    0.0      120.0    [TOOL1       ]
       2 200.0    0.0      255.5    [TOOL2       ]
       3 0.0      0.0      350.0    [TOOL3       ]
       4 -5.5     6.6      10.0     [TOOL4       ]
       5 0.0      0.0      0.0      [            ]
       6 0.0      0.0      0.0      [            ]
       7 0.0      0.0      0.0      [            ]
       8 0.0      0.0      0.0      [            ]
       9 0.0      0.0      0.0      [            ]
      10 0.0      0.0      0.0      [            ]
     Active TOOL $MNUTOOLNUM [G: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 will be displayed.
4 Press F2, "METHOD".
5 Select "Four Point". The tool frame setup / four point screen will be displayed.

Tool frame setup screen (Four Point Method)

```
SETUP Frames
      Tool Frame         Four PT Tool      1/5
      Frame Number: 5
        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
       Approach point 4:      UNINIT
       Mean Error: ******* mm
       Max. Error: ******* mm at point[0]

     Active TOOL $MNUTOOLNUM[G:1] = 1

   [ TYPE ] [METHOD]  FRAME
```

6 Add a comment and teach the approach point. For details, refer to TCP auto set (Three Point Method).

a Press and hold [SHIFT] key and press F5, "RECORD" to record the data of current position as the approach point. As for the taught approach point, "RECORDED" will be displayed.

```
SETUP Frames
      Tool Frame         Four PT Tool      3/5
      Frame Number: 5
        X:    0.0   Y:    0.0   Z:    0.0
        W:    0.0   P:    0.0   R:    0.0
       Comment:                 TOOL5
       Approach point 1:      RECORDED
       Approach point 2:      RECORDED
       Approach point 3:      UNINIT
       Approach point 4:      UNINIT
       Mean Error: ******* mm
       Max. Error: ******* mm at point[0]

     Active TOOL $MNUTOOLNUM[G:1] = 1

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

b When all the approach points are taught, "USED" will be displayed. The tool frame has been set.

```
SETUP Frames
      Tool Frame         Four PT Tool      5/5
      Frame Number: 5
        X:    6.3   Y:    0.8   Z:  311.8
        W:    0.0   P:    0.0   R:    0.0
       Comment:                 TOOL5
       Approach point 1:      USED
       Approach point 2:      USED
       Approach point 3:      USED
       Approach point 4:      USED
       Mean Error:   0.340 mm
       Max. Error:   0.389 mm at point[3]

     Active TOOL $MNUTOOLNUM[G:1] = 1

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

c The "Mean" and "Max." Error represent the deviation of positions of the tool tip at each approach point. The approach point displayed at the line of Max. Error can be thought as the most influential point to inaccuracy of TCP. Please correct the corresponding point iteratively so that these values become as small as possible.

NOTE

Changing the posture of the main axes significantly may leads to inaccuracy of TCP because of arm bending or backrush. In this case, to overcome this, please try teaching approach points, around the area where the tool is actually used, largely rotating the wrist axes only as much as possible.

An example of teaching by largely rotating the wrist axes only:

1. Direct the flange to just under (near 90 degrees for J5) and then record the first approach point.
2. Rotate 90 degrees around J6 axis and jog the robot so that the tool tip touches the common point. Then record the second point.
3. Jog the robot to the first point and rotate 30 degrees around J5 axis. Then jog the robot so that the tool tip touches the common point. Finally record the third point.
4. Jog the robot to the first point and rotate 30 degrees around J4 axis. Then jog the robot so that the tool tip touches the common point. Finally record the last point.

7 Press [PREV] key. The tool frame list screen will be displayed. You can see all the tool frame settings.

```
SETUP Frames
      Tool Frame     / Four PT Tool          5/10
           X        Y         Z       Comment
       1 100.0    0.0      120.0    [TOOL1       ]
       2 200.0    0.0      255.5    [TOOL2       ]
       3 0.0      0.0      350.0    [TOOL3       ]
       4 -5.5     6.6      10.0     [TOOL4       ]
       5 6.3      0.8      311.8    [TOOL5       ]
       6 0.0      0.0      0.0      [            ]
       7 0.0      0.0      0.0      [            ]
       8 0.0      0.0      0.0      [            ]
       9 0.0      0.0      0.0      [            ]
      10 0.0      0.0      0.0      [            ]
     Active TOOL $MNUTOOLNUM [G:1] = 1

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

8 To make the set tool frame effective, press F5, "SETIND", then enter the frame number.

⚠ **CAUTION**

1. If you do not press F5, "SETIND", the tool frame will not be effective.
2. After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

NOTE

To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 ”Moving the Robot by Jog Feed”.

9 To delete the data of the set frame, move the cursor to the desired frame and press F4, "CLEAR". You can select whether to clear its comment too or not.

3.9.2 Setting a User Coordinate System

A user coordinate system is a Cartesian coordinate system defined for each work space by the user. If the coordinate system is not defined, the world coordinate system substitutes for it.
Define the user coordinate system by (x, y, z) indicating the position of the zero point and (w, p, r) indicating the angular displacement around the X-, Y-, and Z-axes on the world coordinate system.
The user coordinate system is used to specify a position register and execute the corresponding position register instruction and position compensation instruction. For the specification of the position register, see Section 7.4, “POSITION REGISTERS.” For the execution of the position register instruction, see Subsection 4.3.2, “Position Data.” For the execution of the position compensation instruction, see Subsection 4.3.5, “Additional Motion Instructions”.

> ⚠ **CAUTION**
> If teaching is made by joint coordinates, changing the user coordinate system does not affect the position variables and position registers. However, note that both position variables and registers are affected by the user coordinate systems when the robot is taught in the Cartesian format.

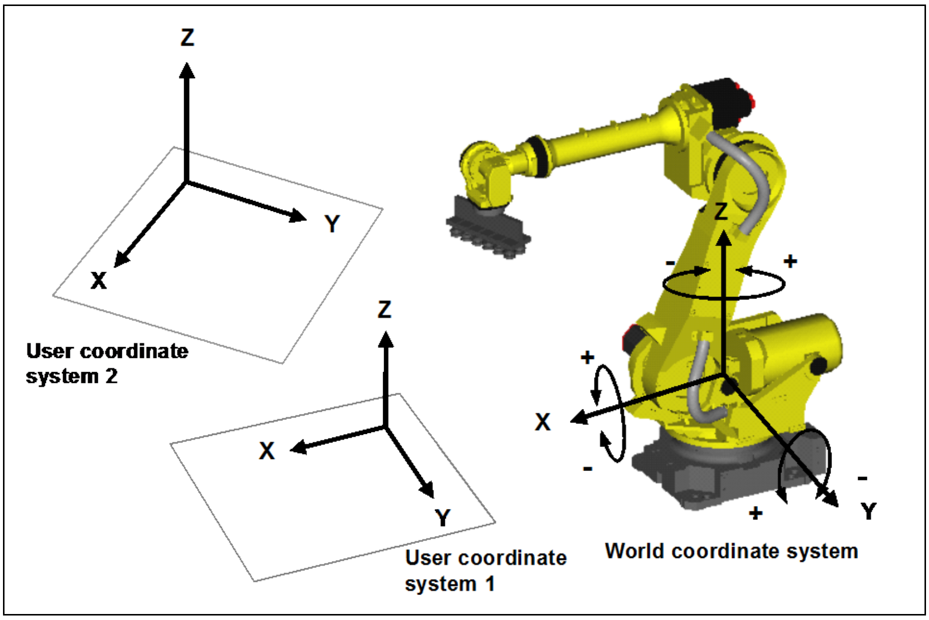


Fig. 3.9.2 (a) World and user coordinate systems

The following system variables are changed by defining the user frame with the frame setup screen. Nine user coordinate systems can be defined. The desired one can be selected.

- $MNUFRAME [group, i] (Frame number i = 1 to 9) is set the value.
- $MNUFRAMENUM [group 1] is set the user frame number you want to use.

The number of the user coordinate systems can be increased up to 253 by the following way. (up to 61 in R-30*i*B and R-30*i*B Mate)

1. Perform a controlled start.
2. Press MENU key.
3. Select “4. Variables”.
4. Set the number of user coordinate systems to the system variable $SCR.$MAXNUMUFRAM. The maximum number of user coordinate systems is 253. (The maximum number of user coordinate systems is 61 in R-30*i*B and R-30*i*B Mate.)
5. Perform a cold start.

NOTE
If you set maximum number of user coordinate system over 61 in R-30*i*B Plus, TP programs saved in the system cannot be load to either R-30*i*B or R-30*i*B Mate.

The user frame can be defined by the following three methods.

Three Point Method

Teach the following three points: the origin of the x-axis, the point which specifies the positive direction of the x-axis, and the point on the x-y plane.

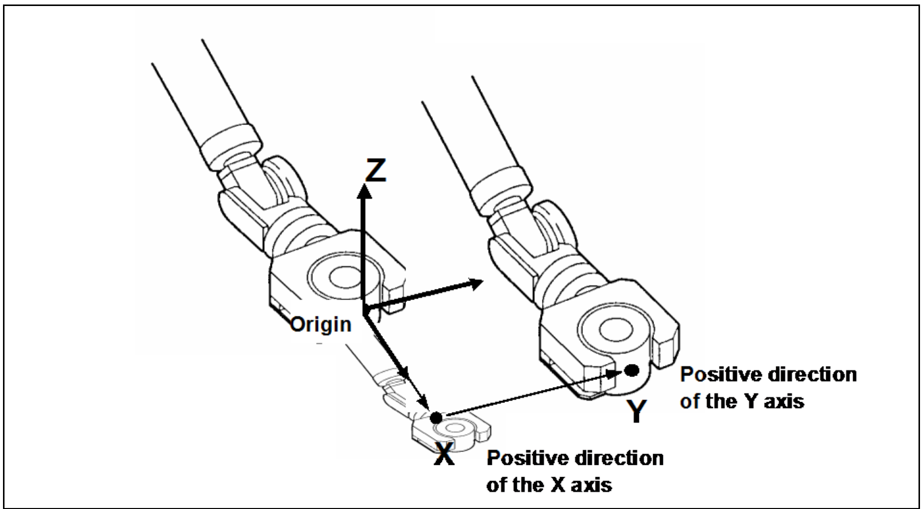


Fig. 3.9.2 (b) Three point method

Four Point Method

Teach the following four points: the origin of the x-axis parallel to the frame, the point which specifies the positive direction of the x-axis, a point on the x-y plane, and the origin of the frame.

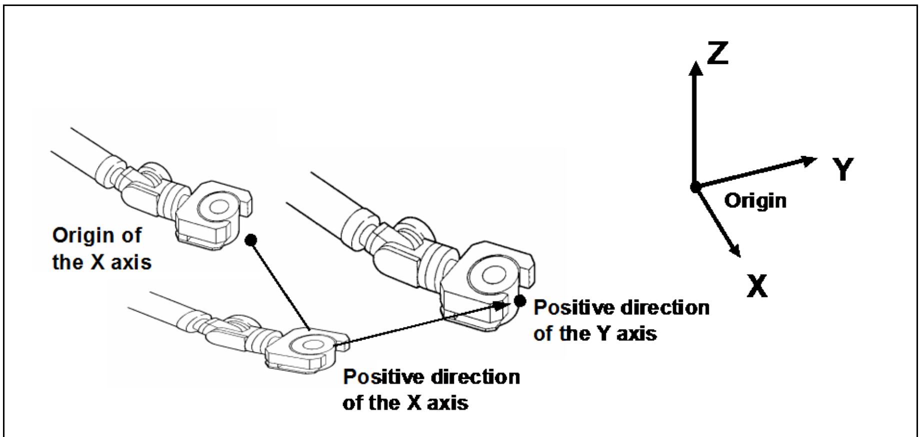


Fig. 3.9.2 (c) Four point method

Direct 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.

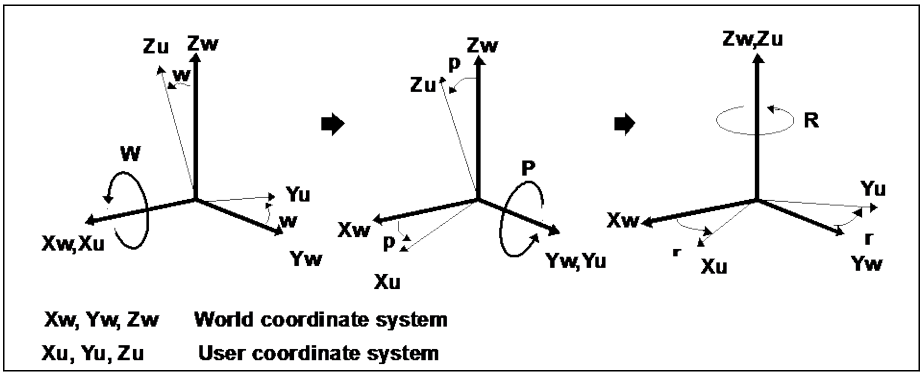


Fig. 3.9.2 (d) Meaning of (w,p,r) used in direct list method

Procedure 3-18 Setting Up User Frame Using Three Point Method

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select “6 SETUP”.
3 Press the F1, [TYPE]. The screen change menu will be displayed.
4 Select "Frames".
5 Press F3, [OTHER] and then select User Frame.
6 The user frame list screen is displayed.

User frame list screen

```
SETUP Frames
      User Frame     / 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[G:1] = 1

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

7 Move the cursor to the line of the user frame number you want to set.
8 Press F2, "DETAIL". The user frame setup screen of the selected frame number is displayed.
9 Press F2, "METHOD" and then select "Three Point".

User frame setup screen (Three Point Method)

```
SETUP Frames
      User Frame         Three Point          1/4
      Frame Number:         1

        X:    0.0   Y:    0.0   Z:    0.0
        W:    0.0   P:    0.0   R:    0.0

       Comment:   *******************
       Orient Origin Point:    UNINIT
       X Direction Point:      UNINIT
       Y Direction Point:      UNINIT


     Active UFRAME $MNUFRAMENUM[G:1] = 0

   [ TYPE ]  [METHOD]   FRAME
```

10 To add a comment:
   a Move the cursor to the comment line and press [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.
11 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 [SHIFT] key and press F5, "RECORD" to record the current position as the approach point. As for the taught reference point, "RECORDED" is displayed.

```
SETUP Frames
      User Frame         Three Point          3/4
      Frame Number: 1

        X:     0.0   Y:     0.0   Z:     0.0
        W:     0.0   P:     0.0   R:     0.0

       Comment:REFERENCE FRAME
       Orient Origin Point:  RECORDED
       X Direction Point:    RECORDED
       Y Direction Point:      UNINIT


     Active UFRAME $MNUFRAMENUM[G: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.

> ⚠ **CAUTION**
> If you are using robots with less than 6 robot axes, such as 4 axes robot and 5 axes robot, please pay attention to the following points.
>
> 1 If positions of a TP program were edited by entering numeric values directly, the positions may become unreachable, and as the result the robot may not be able to move. In this case, please correct the value of rotating angle (w, p, r) by Direct List Method. The user frame may have unintended slopes due to deviation of the reference positions recorded for this method. (E.g. despite your expectation that w, p will be set to zero, these might be set to non-zero value less than 1.0.)
>
> 2 If the user frame was created/changed after creation of a TP program, positions of the TP program may become unreachable, and as the result the robot may not be able to move. In this case, please re-teach by TOUCHUP.

```
SETUP Frames
      User Frame         Three Point          4/4
      Frame Number: 1

        X:     143.6    Y:     0.0    Z:  10.0
        W:       0.123  P:     2.34   R:   3.2

       Comment:          REFERENCE FRAME
       Orient Origin Point:          USED
       X Direction Point:            USED
       Y Direction Point:            USED


     Active UFRAME $MNUFRAMENUM[G:1] = 0

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

12 To move to a recorded position, press and hold [SHIFT] key and press F4, "MOVE_TO".

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

14 To display the user frame list screen, press [PREV] key. You can see the settings for all user frames.

```
SETUP Frames
      User Frame     / Three Point         1/9
             X        Y       Z       Comment
       1 143.6       0.0    10.0 [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 TOOL $MNUFRAMNUM[G:1] = 0

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

15 To make the set user frame effective, press F5, "SETIND", then enter the frame number.

⚠ CAUTION

1. If you do not press F5, "SETIND", the user frame will not be effective.
2. After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

NOTE

To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 ”Moving the Robot by Jog Feed”.

16 To delete the data of the set frame, move the cursor to the desired frame and press F4, "CLEAR".

Procedure 3-19 Setting User Frame Using Four Point Method

Step

1 Display the user frame list screen (Refer to the three point method).

```
SETUP Frames
      User Frame       / Direct Entry          2/9
           X        Y         Z       Comment
       1  143.6      0.0    10.0 [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 $MNUFRAMENUM[G: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 will be displayed.
4 Press F2, "METHOD".
5 Select "Four Point". The user frame setup / four point screen will be displayed.

User frame setup screen (Four Point Method)

```
SETUP Frames
     User Frame          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 $MNUFRAMENUM[G:1] = 1

   [ TYPE ][ METHOD ]  FRAME
```

6 Add a comment and teach the reference point. For details, refer to TCP auto set (Three Point Method).

> ⚠ **CAUTION**
> If you are using robots with less than 6 robot axes, such as 4 axes robot and 5 axes robot, please pay attention to the following points.
>
> 1 If positions of a TP program were edited by entering numeric values directly, the positions may become unreachable, and as the result the robot may not be able to move. In this case, please correct the value of rotating angle (w, p, r) by Direct List Method. The user frame may have unintended slopes due to deviation of the reference positions recorded for this method. (E.g. despite your expectation that w, p will be set to zero, these might be set to non-zero value less than 1.0.)
> 2 If the user frame was created/changed after creation of a TP program, positions of the TP program may become unreachable, and as the result the robot may not be able to move. In this case, please re-teach by TOUCHUP.

```
SETUP Frames
      User Frame         Four Point           5/5
      Frame Number: 2

       X:  143.6    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 FRAME $MNUFRAMENUM[G:1] = 1

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

7 Press [PREV] key. The user frame list screen is displayed. You can see all the user frame settings.

```
SETUP Frames
      User Frame        / Four Point          2/9
            X        Y         Z       Comment
       1 143.6      0.0    10.0 [REFERENCE FR> ]
       2 143.6    525.5    43.9 [RIGHT FRME    ]
       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 $MNUFRAMENUM[G:1] = 1

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

8 To make the set user frame effective, press F5, "SETIND", then enter the frame number.

⚠ CAUTION

1 If you do not press F5, "SETIND", the user frame will not be effective.
2 After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

NOTE

To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 ”Moving the Robot by Jog Feed”.

9 To delete the data of the set frame, move the cursor to the desired frame and press F4, "CLEAR".

Procedure 3-20 Setting User Frame Using Direct List Method

Step

1 Display the user frame list screen (Refer to the three point method).

```
SETUP Frames
      User Frame        / Direct Entry         3/9
            X        Y         Z        Comment
       1 143.6       0.0    10.0 [REFERENCE FR> ]
       2 143.6     525.5    43.9 [RIGHT FRME    ]
       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 $MNUFRAMENUM[G: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 [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)

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

     Active UFRAME $MNUFRAMENUM[G:1] = 1

   [ TYPE ] [METHOD]  FRAME
```

6 Add a comment and enter the coordinate values. For details, refer to tool frame (Direct List Method).

```
SETUP Frames
      User Frame         Direct Entry          4/7
      Frame Number:        3
       1  Comment:        LEFT FRAME
       2 X:                     143.600
       3 Y:                    -525.500
       4 Z:                      43.900
       5 W:                       0.123
       6 P:                       2.340
       7 R:                       3.200
         Configuration      N D B, 0, 0, 0

     Active UFRAME $MNUFRAMENUM[G:1] = 1

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

7 To display the user frame list screen, press [PREV] key. You can see the settings of all the user frame.

```
SETUP Frames
      User Frame        / Direct Entry         3/9
            X        Y         Z        Comment
       1 143.6       0.0    10.0 [REFERENCE FR> ]
       2 143.6     525.5    43.9 [RIGHT FRAME   ]
       3 143.6    -525.5    43.9 [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 $MNUFRAMENUM[G:1] = 1

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

8 To get the set user frame as effective, press F5, "SETIND".

> ⚠ **CAUTION**
> 1 If you do not press F5, "SETIND", the user frame will not be effective.
> 2 After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

> **NOTE**
> To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 "Moving the Robot by Jog Feed".

9 To delete the data of the set frame, move the cursor to the desired frame and press F4, CLEAR.

Procedure 3-21 Method to change User Frame Number into "0"(World Frame)

Step

1 Display the user frame list screen.

```
SETUP Frames
      User Frame        / Direct Entry         3/9
            X        Y         Z        Comment
       1 143.6       0.0    10.0 [REFERENCE FR> ]
       2 143.6     525.5    43.9 [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 $MNUFRAMENUM[G:1] = 1

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

2 Press the [NEXT] key.
3 Press F2, "CLRIND". Then, $MNUFRAMENUM[x] will be changed into 0.

```
     Active UFRAME $MNUFRAMENUM[G:1] = 0

   [ TYPE ]  CLRIND
```

3.9.3 Setting a Jog Coordinate System

A jog coordinate system is a Cartesian coordinate system defined in a work space by the user. It is used to efficiently move the robot by Cartesian jog in the work space. (See Subsection 5.2.3.)
The jog coordinate system is defined by (x, y, z) indicating the position of the zero point, and (w, p, r) indicating the angular displacement around the X-, Y-, and Z-axes on the world coordinate system.

NOTE
You need not take care of the jog frame origin, since it is used only when the jog frame is selected as the manual-feed coordinate system. The zero point of the jog coordinate system has no special meaning. Select any convenient position for defining the jog coordinate system.

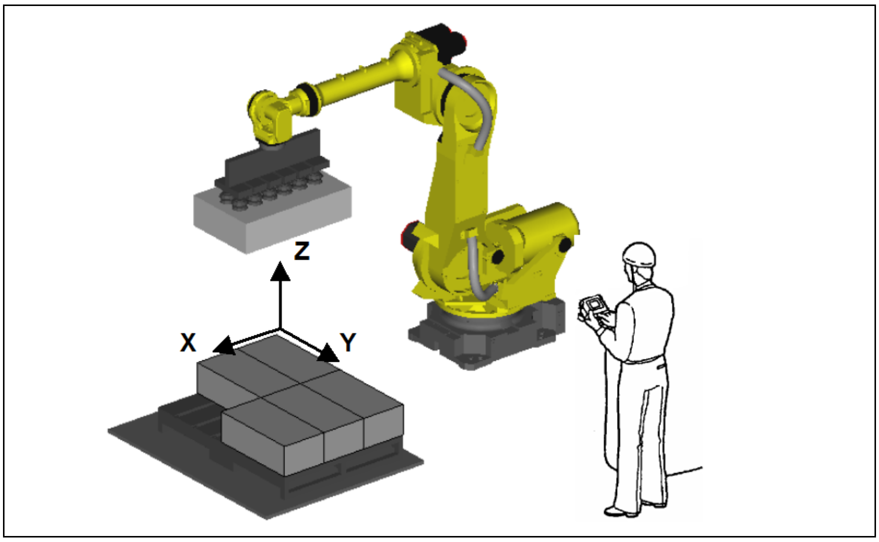


Fig. 3.9.3 Jog coordinate system

The following system variables are changed by setting the jog frame with the frame setup screen.

- $JOG_GROUP [group] .$JOGFRAME is set the jog frame you want to use.

Five jog frames can be set and they can be switched according to the situation. It is substituted by the world frame when undefined.
Jog frame can be set by two methods.

Three Point Method
Three reference points need be taught. They are the start point of the X-axis, the positive direction of the X-axis, and one point on the X-Y plane. The start point of the X-axis is used as the origin of the frame. Refer to Fig. 3.9.2 (b).

Direct 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 Fig. 3.9.2 (d).

Procedure 3-22 Setting Up Jog Frame Using Three Point Method

Step
1 Press [MENU] key. The screen menu will be displayed.
2 Select “6 SETUP”.
3 Press the F1, [TYPE]. The screen change menu will be displayed.
4 Select "Frames".

5 Press F3, [OTHER].
6 Select "Jog Frame". Jog frame entry screen will be displayed.

Jog frame list screen

```
SETUP Frames
      Jog Frame       / 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 JOG FRAME[G:1] = 1

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

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 will be displayed.
9 Press F2, "METHOD".
10 Select "Three Point".

Jog frame setup screen (three point method)

```
SETUP Frames
      Jog Frame         Three Point           1/4
      Frame Number: 1

        X:    0.0   Y:    0.0   Z:    0.0
        W:    0.0   P:    0.0   R:    0.0

       Comment:*******************
       Orient Origin Point:    UNINIT
       X Direction Point:      UNINIT
       Y Direction Point:      UNINIT

     Active JOG FRAME[G:1] = 1

   [ TYPE ]  [METHOD]  FRAME
```

11 Add a comment and teach the reference point. For details, refer to TCP auto set (Three Point Method).

```
SETUP Frames
      Jog Frame         Three Point           4/4
      Frame Number: 1

        X:  143.6     Y:  525.5    Z:   60.0
        W:    0.123   P:    2.34   R:    3.2

       Comment:WORK AREA 1
       Orient Origin Point:      USED
       X Direction Point:        USED
       Y Direction Point:        USED

     Active JOG FRAME[G:1] = 1

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

12 Press [PREV] key. The jog frame list screen will be displayed. You can see all the jog frame settings.

```
SETUP Frames
      Jog Frame        / Three Point           1/5
            X        Y         Z       Comment
        1   143.6    525.5     60.0  [Work Area 1 ]
        2     0.0      0.0      0.0  [            ]
        3     0.0      0.0      0.0  [            ]
        4     0.0      0.0      0.0  [            ]
        5     0.0      0.0      0.0  [            ]

     Active JOG FRAME[G:1] = 1

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

13 To make the set jog frame effective, press F5, "SETIND", then enter the frame number.

> ⚠ **CAUTION**
> 1 If you do not press F5, "SETIND", the jog frame will not be effective.
> 2 After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

> **NOTE**
> To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 ”Moving the Robot by Jog Feed”.

14 To delete the data of the set frame, move the cursor to the desired frame and press F4, CLEAR.

Procedure 3-23 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
      Jog Frame         / Direct Entry       2/5
            X        Y         Z       Comment
        1   143.6    525.5     60.0  [Work Area 1 ]
        2     0.0      0.0      0.0  [            ]
        3     0.0      0.0      0.0  [            ]
        4     0.0      0.0      0.0  [            ]
        5     0.0      0.0      0.0  [            ]

     Active JOG FRAME[G:1] = 1

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

2 Move the cursor to the jog frame number line you want to set.
3 Press F2, "DETAIL" or press [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
      Jog Frame        Direct Entry          1/7
      Frame Number:      2
       1  Comment:       ********************
       2  X:                        0.000
       3  Y:                        0.000
       4  Z:                        0.000
       5  W:                        0.000
       6  P:                        0.000
       7  R:                        0.000
          Configuration:      N D B, 0, 0, 0

     Active JOG FRAME[G:1] = 1

   [ TYPE ] [METHOD]  FRAME
```

6 Add a comment and teach the reference point. For details, refer to TCP auto set (Three Point Method).

```
SETUP Frames
      Jog Frame         Direct Entry         4/7
      Frame Number:      3
       1  Comment:                WORK AREA 2
       2  X:                      103.600
       3  Y:                     -236.000
       4  Z:                       90.000
       5  W:                        0.000
       6  P:                        0.000
       7  R:                        0.000
          Configuration:      N D B, 0, 0, 0

     Active JOG FRAME[G:1] = 1

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

7 Press [PREV] key. The jog frame list screen will be displayed. You can see all the jog frame settings.

```
SETUP Frames
      Jog Frame        / Direct Entry       2/5
            X        Y         Z       Comment
       1   143.6    525.5     60.0  [Work Area 1 ]
       2  1003.0   -236.0     90.0  [Work Area 2 ]
       3     0.0      0.0      0.0  [            ]
       4     0.0      0.0      0.0  [            ]
       5     0.0      0.0      0.0  [            ]




     Active JOG FRAME[G:1] = 1

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

8 To make the set jog frame effective, press F5, "SETIND", then enter the frame number.

⚠ CAUTION

1 If you do not press F5, "SETIND", the jog frame will not be effective.
2 After all coordinate systems are set, the setting information should be saved in external storage in case the information needs to be re-loaded. Otherwise, the current setting information would be lost when it is changed.

> **NOTE**
> To select the number of a coordinate system to be used, the jog menu can also be used. See Subsection 5.2.3 ”Moving the Robot by Jog Feed”.

9 To delete the data of the set frame, move the cursor to the desired frame and press F4, "CLEAR".

3.9.4 Setting a Cell Coordinate System

In 4D graphic function, the cell coordinate system is used to describe the robot position in the work cell. By setting the cell coordinate system, the relation of the position between the robots can be described.

The cell coordinate system can be set by the following two methods.

Direct List Method

The origin position x, y and z of the world frame in the cell frame and the rotating angle w, p, and r around the X-,Y-,and Z-axis of the cell frame can be input directly.

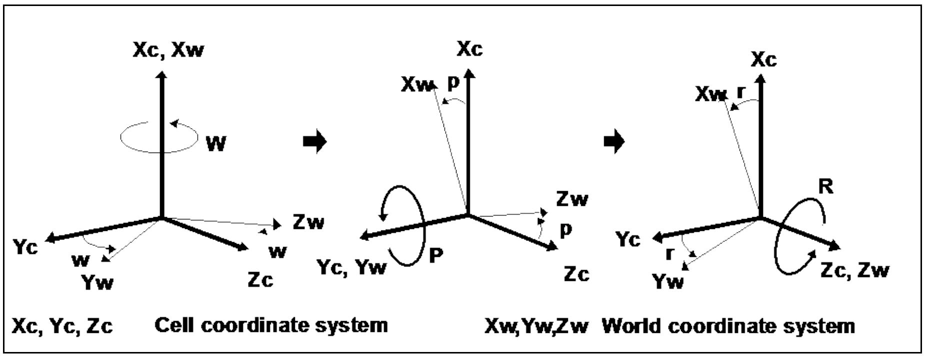


Fig. 3.9.4 Meaning of (w, p, r) used in direct list method

Copy from other coordinate system

The cell coordinate system can be set by copy from the user frame or CD PAIR.

Procedure 3-24 Setting Up Cell Frame Using the Direct Entry Method

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select “6 SETUP”.
3 Press F1, [TYPE]. The screen change menu will be displayed.
4 Select "Frames".
5 Press F3, [OTHER].
6 Select "Cell Frame". The cell frame list screen will be displayed.

```
SETUP Frames
       Cell Frame        /

       Frame Status:       Not Set
       Calibration Type:   Not Set

        X:    0.0  Y:    0.0  Z:    0.0
        W:    0.0  P:    0.0  R:    0.0




   [ TYPE ]  DETAIL  [OTHER]  CLEAR   COPY
```

7 Press F2, "DETAIL" or press [ENTER] key.
8 The cell frame setup screen using the direct entry method is displayed.

Cell frame setup screen (direct entry method)

```
SETUP Frames
       Cell Frame     / Direct Entry         1/6

       Frame Status:       Not Set
       Calibration Type:   Not Set

      1 X:                           0.0
      2 Y:                           0.0
      3 Z:                           0.0
      4 W:                           0.0
      5 P:                           0.0
      6 R:                           0.0

   [ TYPE ]                          APPLY
```

9 Enter the coordinate value of the cell frame.
   a Move the cursor to each component.
   b Enter new numerical value by using numerical key.
   c Press [ENTER] key. A new numerical value is set.

```
SETUP Frames
       Cell Frame     / Direct Entry         1/6

       Frame Status:       Not Set
       Calibration Type:   Not Set

      1 X:                           0.0
      2 Y:                           0.0
      3 Z:                         350.0
      4 W:                         180.0
      5 P:                           0.0
      6 R:                           0.0

   [ TYPE ]                          APPLY
```

10 Press F5, APPLY. The cell frame list screen will be displayed. The data of “Frame Status” is changed to “Calibrated”, and the data of “Calibration Type” is changed to “Direct Entry”.

```
SETUP Frames
       Cell Frame        /

       Frame Status:       Calibrated
       Calibration Type:   Direct Entry

        X:    0.0  Y:    0.0  Z:  350.0
        W:  180.0  P:    0.0  R:    0.0

   [ TYPE ]  DETAIL  [OTHER]  CLEAR   COPY
```

11 To delete the data of the set frame, press F4, "CLEAR".

Procedure 3-25 Copying Cell Frame

Step

1 Display the cell frame list screen. (Refer to the direct entry method.)

```
SETUP Frames
       Cell Frame        /

       Frame Status:       Not Set
       Calibration Type:   Not Set

        X:     0.0  Y:     0.0  Z:     0.0
        W:     0.0  P:     0.0  R:     0.0

   [ TYPE ]  DETAIL  [OTHER]  CLEAR   COPY
```

2 Press F5, "COPY". The cell frame copy screen is displayed.

```
SETUP Frames
       Cell Frame       / Copy Cell Frame       1/3

         Source:           CELL FRAME

         Destination::     UFRAME

         Frame Index:       1

   [ TYPE ]                 [CHOICE]   APPLY
```

3 Enter Source and Destination.

4 User frame or cell frame or CD PAIR can be entered in Source and Destination. (The CD PAIR can be selected only when the coordinate motion option is loaded.)

> **NOTE**
> The same frame can not be selected in both Source and Destination.
> Either Source or Destination must be set the cell frame.
> In case that the cell frame is copied from the CD PAIR, the selected source CD PAIR should already be calibrated. Also, the cell frame of the follower group of the CD PAIR should be calibrated.

3.9.5 Setting a Cell Floor

The cell floor is the coordinate system to describe the floor in which the robot is placed in 4D graphics function.

To set the cell floor, set the position and the attitude of the floor in the cell coordinate system.

Set the cell floor according to Procedure 3-26.

Procedure 3-26 Setting Up Cell Floor

Step

1 Press [MENU] key. The screen menu will be displayed.

2 Select “6 SETUP”.

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

4 Select "Frames".
5 Press F3, [OTHER].
6 Select "Cell Floor". The cell floor list screen will be displayed.

```
SETUP Frames
       Cell Floor         /                    1/2

        X:    0.0     Y:    0.0      Z:        0.0
        W:    0.0     P:    0.0      R:        0.0




  [ TYPE ]          [ OTHER ]
```

7 Enter the coordinate value of the cell floor.
   a Move the cursor to each component.
   b Enter new numerical value by using numerical key.
   c Press [ENTER] key. A new numerical value is set.

```
SETUP Frames
       Cell Floor         /                    1/2

        X:    0.0     Y:    50.0     Z:        0.0
        W:    0.0     P:    90.0     R:        0.0




  [ TYPE ]          [ OTHER ]
```

8 After the cell floor is set, when the displayed screen is changed, new cell floor is available.

3.10 SETTING A REFERENCE POSITION

A reference position is a fixed (predetermined) position that is frequently used in a program or when the robot is moved by jog feed. The reference position is a safe position, which is usually distant from the operating area of the machine tool or peripheral equipment. Up to 10 reference positions can be defined. Settings of reference positions are saved to SYSVARS.SV.

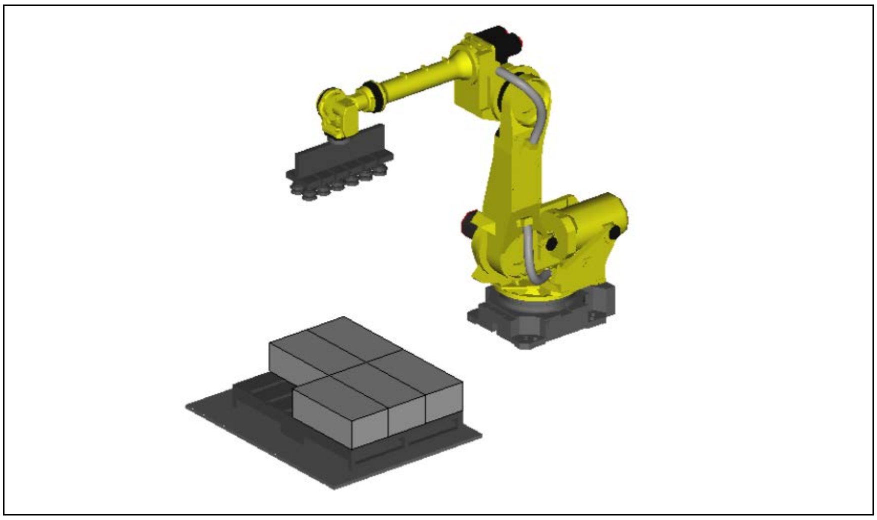


Fig. 3.10 Reference position

When the robot is at the reference position, a predetermined digital signals, DO, is output. If the reference position is invalidated, the DO signal is not output.
When the robot is at reference position 1, the reference position output signal (ATPERCH) of the peripheral device I/O is output.
For this function, the reference position settings can be disabled so that the signal is not output.
To make the robot move to the reference position, make the program which specifies the return path and execute this program. At this time, also specify the order in which axes returns to the reference position in the program. Moreover, it is convenient to set the return program as a macro instruction. (See Section 9.1, ”MACRO INSTRUCTION”.)
Specify the reference position on the reference position setting screen [6 SETUP Ref Position].

Procedure 3-27 Setting a reference position

Step

1 Press [MENU] key.
2 Select "6. SETUP".
3 Press F1, [TYPE].
4 Select "Ref Position". The reference position selection screen will be displayed.

Reference position selection screen

```
REF POSN
                                                 1/10
      No.   Enb/Dsbl  @Pos    Comment
       1    DISABLE   FALSE [                   ]
       2    DISABLE   FALSE [                   ]
       3    DISABLE   FALSE [                   ]
       4    DISABLE   FALSE [                   ]
       5    DISABLE   FALSE [                   ]
       6    DISABLE   FALSE [                   ]
       7    DISABLE   FALSE [                   ]
       8    DISABLE   FALSE [                   ]
       9    DISABLE   FALSE [                   ]
      10    DISABLE   FALSE [                   ]

   [ TYPE ]           DETAIL    ENABLE    DISABLE
```

5 Press F3, "DETAIL". The detailed reference position screen will be displayed.

Reference position detail screen

```
REF POSN
      Reference Position                          1/13
           Ref.Position Number:                      1
        1  Comment                    [**************]
        2  Enable/Disable:              DISABLE
        3  Is a valid HOME:             DISABLE
        4  Signal definition:           DO [    0]
        5    J1:      0.000  +/-      0.000
        6    J2:      0.000  +/-      0.000
        7    J3:      0.000  +/-      0.000
        8    J4:      0.000  +/-      0.000
        9    J5:      0.000  +/-      0.000
       10    J6:      0.000  +/-      0.000

   [ TYPE ]                             RECORD
```

6 To enter a comment, follow these steps:
  a Place the cursor on the comment line and press [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 [ENTER] key.

7 In the “Signal definition” line, specify the digital output signal to be output when the tool is at the reference position.

NOTE
Do not set same signal as other reference position to Signal Definition. If same signal is set to Signal Definition in the setup screens of two or more reference positions, unpredictable results would be produced. For example, the signal set in Signal Definition is not output even if the robot is in the reference position.

```
REF POSN

      4  Signal definition:        RO [    0]

 [ TYPE ]               DO        RO
```

```
REF POSN

      4  Signal definition:        RO [    1]

 [ TYPE ]
```

8 To teach the reference position, place the cursor on the setting fields J1 to J9. While pressing [SHIFT] key, press F5, "RECORD". The current position is recorded as the reference position.

9 To enter the numeric value of the reference position directly, place the cursor on the setting fields J1 to J9 and enter the coordinates of the reference position. Enter the coordinates in the left column and allowable errors in the right column. Moreover, the value entered to the setting field which specifies an unused axis is ignored.

```
REF POSN
      Reference Position                     1/13
          Ref.Position Number:                  1
       1  Comment                 [Refpos1      ]
       2  Enable/Disable:           ENABLE
       3  Is a valid HOME:          FALSE
       4  Signal definition:        RO [    1]
       5    J1:    129.000  +/-      2.000
       6    J2:    -31.560  +/-      2.000
       7    J3:      3.320  +/-      2.000
       8    J4:    179.240  +/-      2.000
       9    J5:      1.620  +/-      2.000
      10    J6:     33.000  +/-      2.000

 [ TYPE ]                         RECORD
```

10 After the reference position is specified, press PREV key. The reference position selection screen will be displayed again.

```
REF POSN
                                          1/10
      No.   Enb/Dsbl  @Pos    Comment
       1    DISABLE   FALSE [Refpos1           ]
       2    DISABLE   FALSE [                  ]
       3    DISABLE   FALSE [                  ]
       4    DISABLE   FALSE [                  ]
       5    DISABLE   FALSE [                  ]
       6    DISABLE   FALSE [                  ]
       7    DISABLE   FALSE [                  ]
       8    DISABLE   FALSE [                  ]
       9    DISABLE   FALSE [                  ]
      10    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
                                          1/10
      No.   Enb/Dsbl  @Pos    Comment
       1    ENABLE    FALSE [Refpos1           ]

  [ TYPE ]          DETAIL   ENABLE   DISABLE
```

3.11 JOINT OPERATING AREA

The software restricts the operating area of the robot according to a specified joint operating area. The standard operating area of the robot can be changed by specifying the joint operating area.
Specify the joint operating area at [6 SYSTEM Axis Limits] on the joint operating area setting screen.

> ⚠ **WARNING**
> 1 The robot operating area should not be controlled only by the joint moving range function. Limit switches and mechanical stoppers should be used together with the function. Otherwise, injury or property damage would occur.
> 2 The mechanical stoppers should be adjusted to the software settings. Otherwise, injury or property damage would occur.

> ⚠ **CAUTION**
> Changing the joint 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, cycle power of the controller to enable the new setting.

Procedure 3-28 Setting the joint operating area

Step

1 Press MENU key. The screen menu will be displayed.
2 Select "6 SYSTEM".
3 Press F1, [TYPE]. The screen change menu will be displayed.
4 Select "Axis Limits". The joint operating area setting screen will be displayed.

Joint operating area setting screen

```
SYSTEM Axis Limits
      AXIS     GROUP       LOWER       UPPER   1/56

       1         1       -150.00      150.00  dg
       2         1        -60.00      100.00  dg
       3         1       -110.00       50.00  dg
       4         1       -240.00      240.00  dg
       5         1       -120.00      120.00  dg
       6         1       -450.00      450.00  dg
       7         1          0.00      150.00  dg
       8         1          0.00        0.00  mm
       9         1          0.00        0.00  mm
      10         1          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
A setting of 0.000 indicate that the robot does not have the corresponding axis.

5 Place the cursor on the target axis limits field, and enter a new value from the teach pendant.

```
SYSTEM Axis Limits
      AXIS     GROUP       LOWER       UPPER   2/56

       2         1        -50.00      100.00  dg

   [ TYPE ]
```

⚠ **WARNING**
The axis range cannot be extended beyond the value of factory-set up.

6 Repeat the above step for all the axes.
7 To make the set information effective, turn the controller off and on again in cold start mode (Subsection 5.2.1).

⚠ **WARNING**
Cycling power is required to make a new setting valid. Otherwise, injury or property damage would occur.

3.12 USER ALARM

In the user alarm setup screen, the message displayed when the user alarm is generated is set.
The user alarm is the alarm which is generated when the user alarm instruction is executed. (See Subsection 4.15.2 ”User Alarm Instruction”.)
Settings for user alarm is done in the user alarm setup screen [6 SETUP. User Alarm].

Procedure 3-29 Setting Up the User Alarm

Step

1 Select [MENU] key. The screen menu will be displayed.
2 Select "6 SETUP".
3 Press F1, [TYPE]. The screen change menu will be displayed.
4 Select User Alarm. The user alarm setup screen will be displayed.

User alarm setup screen

```
Setting/User Alarm
                                         1/10
      Alarm No.           User Message
       [1]:    [                                ]
       [2]:    [                                ]
       [3]:    [                                ]
       [4]:    [                                ]
       [5]:    [                                ]
       [6]:    [                                ]
       [7]:    [                                ]
       [8]:    [                                ]
       [9]:    [                                ]
       [10]:   [                                ]

   [ TYPE ]
```

5 Move the cursor to the line of the user alarm number you want to set and press [ENTER] key. Enter the message with the function keys.

```
Setting/User Alarm
                                         3/10
      Alarm No.           User Message
       [1]:    [                                ]
       [2]:    [                                ]
       [3]:    [WORK                            ]
       [4]:    [                                ]
       [5]:    [                                ]
       [6]:    [                                ]
       [7]:    [                 Alpha input 1
       [8]:    [                   Upper Case
       [9]:    [                   Lower Case
       [10]:   [                   Punctuation
     Old Value:                    Options

   ABCDEF   GHIJKL   MNOPRQ   STUVWX   YZ_@*.
```

6 When you are finished to input the message of the user alarm, press [ENTER] key.
The user alarm message has been set.

```
Setting/User Alarm
                                         3/10
      Alarm No.          User Message
       [1]:    [                                 ]
       [2]:    [                                 ]
       [3]:    [NO WORK                          ]
       [4]:    [                                 ]
       [5]:    [                                 ]
       [6]:    [                                 ]
       [7]:    [                                 ]
       [8]:    [                                 ]
       [9]:    [                                 ]
       [10]:   [                                 ]

   [ TYPE ]
```

3.13 VARIABLE AXIS AREAS

On the variable axis area setting screen, multiple (up to three) sets of stroke limits can be set for the J1 axis and an additional axis.
The variable axis area function allows the user to switch from one set of stroke limits to another during program execution.
* This function is offered by the specific robot only.

Upper limit

Indicates the upper limit for a joint operating area. Operating area in the plus direction.

Lower limit

Indicates the lower limit for a joint operating area. Operating area in the minus direction.

After changing an upper or lower limit, turn off the power to the controller and then turn it on with a cold start. With a cold start, the new upper or lower limit takes effect and the selected joint operating area is returned to the standard value ($PARAM_GROUP.$SLMT_**_NUM=0).

> ⚠ **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 3-30 Setting a variable axis area

Step

1 Press [MENU] key. The screen menu appears.
2 Select "6 SETUP".
3 Press F1, [TYPE]. The screen switching menu appears.
4 Select "Stroke limit". The variable axis area setting screen appears.

Variable axis setting screen

```
Stroke limit setup
                                                  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:
      $MMR_GRP[1].$SLMT_J1_NUM = 0



   [ TYPE ]  GROUP#   AXIS#
```

5 Position the cursor to the desired axis area. Enter new values using the numeric keys on the teach pendant.
- The upper and lower limits must be within the stroke limits of the system. (→Section 3.11, ” JOINT OPERATING AREA”). If an attempt is made to set a value outside the limits, the upper or lower limit is fixed to the system default value.
- To switch from one motion group to another, use the F2 key (group #).
- To set an additional axis, press the F3 key (axis #) to switch to the additional axis setting screen.

6 To make the settings effective, cycle power . When the power is turned on for the first time after the settings have been changed, a cold start is automatically performed.

Procedure 3-31 Using a variable axis area

Condition

■ A proper axis area has been set and is effective.

Step

To switch to the joint operating area that has been set on the variable axis area setting screen during program execution, use the parameter instruction (→ Subsection 4.15.8, ”Parameter Instruction”). For example, after the following program has been executed:

```
PRG1
                                                  1/3
        1:   $MRR_GRP[1].$SLMT_J1_NUM=1
        2:   $PARAM_GROUP[1].$SLMT_J1_NUM=1
      [End]

   [ 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
                                                  1/3
        1:   $MRR_GRP[1].$SLMT_E1_NUM=2
        2:   $PARAM_GROUP[1].$SLMT_E1_NUM=2
      [End]

   [ INST ]                           [ EDCMD ]  >
```

3.14 INTERFERENCE PREVENTION AREA FUNCTION

The interference prevention area function is a function that automatically stops the robot when a move instruction that causes the robot to enter the preset interference area is issued, the robot will stop if another robot or peripheral device is located in that interference area and, after confirming that the other robot or peripheral device has moved out of the interference area, automatically releases the robot from the stopped state to restart its operation.
Communication between a robot and a peripheral device is accomplished with a set of interlock signals (one signal for each of input and output). One set of interlock signals is allocated to one interference area. Up to ten interference areas can be defined. (From software version 7DC1/07, 7DD0/01, 7DC2/01, maximum definable number of interference area have been changed from three to ten.)
The relationship between the interlock signals and the robot is as described below.

Output signal

The output signal is off when the tool center point is located inside the interface area. It is on when it is located outside the area.

State	Output signal
Safe (outside the interference area)	On
Dangerous (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 center point enters the interference area, so that the robot actually stops at a position inside the interface area. The faster the operating speed of the robot, the deeper the robot enters the interference area. Consider this and other factors, such as the tool size, to ensure that a sufficiently large interface area is set.

To set up the interference prevention area function, use the [6 SETUP Space function].

To set up the following items, use the "Rectangular Space/DETAILED" screen.

Table 3.14 (a) Items of the Interference prevention area function (area details screen)

Item	Description
Enable/disable	Enables or 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.

<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td>Priority</td><td>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 so please be careful.

2 Check that there are no obstacles such as collision objects and persons around the robots.
3 Disable this function.
4 Move either robot out of the interference area, using a jog operation.</td></tr>
<tr><td>inside/outside</td><td>Specifies whether the inside or outside of a rectangular parallelepiped is to be an interference area.</td></tr>
</table>

To set up the following items, use the "Rectangular Space/SPACE SETUP" screen.

Table 3.14 (b) Items of the Interference prevention area function (area setting screen)

Item	Description
BASIS VERTEX	Position of the vertex of a rectangular parallelepiped 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 parallelepiped must be parallel to the respective axes of the user coordinate system.) If SECOND VERTEX is selected, the rectangular parallelepiped having the reference vertex and the diagonal vertex, specified here, becomes an interference area.

Procedure 3-32 Setting the interference prevention area function

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select "6 SETUP".
3 Press F1, [TYPE]. The screen switching menu will be displayed.
4 Select "Space fnct". The area list screen will be displayed.

Area list screen

```
Rectangular Space
     LIST SCREEN                                 1/3
      No. Enb/Dsbl    Comment        Usage
       1 ENABLE   [            ]Common Space
       2 DISABLE  [            ]Common Space
       3 DISABLE  [            ]Common Space
       4 DISABLE  [            ]Common Space
       5 DISABLE  [            ]Common Space
       6 DISABLE  [            ]Common Space
       7 DISABLE  [            ]Common Space
       8 DISABLE  [            ]Common Space
       9 DISABLE  [            ]Common Space
      10 DISABLE  [            ]Common Space

   [ TYPE ]   GROUP   DETAIL   ENABLE   DISABLE
```

5 The area list screen allows the user to enable and disable each interface area with the function key. To enter a comment, use the procedure below:
   a. Move the cursor to the desired comment line and press [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. Press [ENTER] key.
6 To set up an item other than Enb/Dsbl or Comment, press F3, "DETAIL". The detailed screen will be displayed.

Area detailed screen

```
 Rectangular Space
     DETAILED SCREEN                             1/7

          SPACE   :1        GROUP:1
          USAGE   :Common Space
       1  Enable/Disable:          DISABLE
       2  Comment:            [            ]
       3  Output Signal:         DO [  0]
       4  Input Signal:          DI [  0]
       5  Priority:              High
       6  Inside/Outside:        Inside
       7  Common Space Num:        0

   [ TYPE ]   SPACE   SETUP   ENABLE   DISABLE
```

7 Position the cursor to the desired item. Change the setting of the item using the function or numeric keys.
8 To set an area, press F2, "SPACE". The space setup screen will be displayed.

Space setup screen

```
Rec
     SPACE SETUP                            1/4

          SPACE   :1        GROUP:1

         UFRAME: 0         UTOOL: 1
      1 :  BASIS VERTEX       [SIDE LENGTH  ]
      2 :X 0.0    mm     0.0      mm
      3 :Y 0.0    mm     0.0      mm
      4 :Z 0.0    mm     0.0      mm

   [ TYPE ]  OTHER                   RECORD
```

9 The reference vertex and the side lengths or diagonal vertex can be set in either of two ways:

a. Position the cursor to the X, Y, and Z coordinate fields and enter the desired coordinates directly using the numeric keys.

b. Move the robot to a vertex of a rectangular parallelepiped, then read the current position of the robot with [SHIFT] key +F5, "RECORD".

NOTE

1 If UF or UT is to be changed, perform operation b first. By means of this operation, the current UF or UT value is selected.

2 When the user coordinate system values are changed, the spatial position of the interference area does not change. When the user coordinate system values have been changed and an interference area is to be defined in the new user coordinate system, use [SHIFT] key +F5, "RECORD" to set an interference area again.

10 After setting the area, press PREV key. The area detailed screen reappears. To return to the area list screen, press PREV key again.

3.15 SYSTEM CONFIG MENU

The System Config Menu includes some important components which should be set when the system is established.
In the system config menu, the following items can be set.

- Use HOT START
- I/O power fail recovery
- HOT START / COLD START autoexec program
- HOT START done signal
- Restore selected program
- Enable UI signals
- START for CONTINUE only
- CSTOPI for ABORT
- Abort all programs by CSTOPI
- PROD-START depend on PNSTROBE
- Detect FAULT_RESET signal
- USE PPABN signal
- WAIT timeout
- RECEIVE timeout
- Return to top of program
- Original program name (F1 to F5)
- Default logical command
- Maximum/Minimum of ACC instruction
- WJNT for default motion
- Auto display of alarm menu
- Force Message
- Reset CHAIN FAILURE detection
- Allow force I/O in AUTO mode
- Allow chg.ovrd. in AUTO mode
- Signal to set in AUTO mode
- Signal to set in T1 mode
- Signal to set in T2 mode
- Signal to set if E-STOP
- Simulated status signal/Simulated input wait time
- Setting to be assumed when the simulated signal skip function is enabled
- Set when prompt displayed
- Output when WAIT on Input
- Signal to be output at an override of 100%
- Hand broken
- Remote / Local setup
- UOP auto assignment
- Multiple-program selection
- WAIT at Taught Position
- Brake control ECO mode
- J7, J8 jog key Setup
- Collection name
- Confirmation of TOUCHUP
- No motion PR operate mode
- Use No-display Sub-Program
- Export diagnosis data

- Check Sim. I/O when FWD/BWD
- Confirmation for AUTO
- Ambient Temperature
- Allow I/O chg. in servo-off
- Prohibit I/O change anytime
- Incremental Jog
- Zero 0% Override

Table 3.15 System config menu

Items	Descriptions
Use HOT START (Hot Start)	When the hot start is set to TRUE, hot start is done at turning on the controller. (Default setting = TRUE)
I/O power fail recovery	Specifies whether or how to perform I/O power failure recovery if the hot start function is enabled and how to perform simulated recovery if the hot start function is disabled. There are four power failure recovery modes, as described below. - NOT RECOVER I/O power failure recovery is not performed regardless of whether the hot start function is enabled. All outputs are turned off, and the simulated state is reset. - RECOVER SIM Simulated-state recovery is performed regardless of whether the hot start function is enabled, and the simulated state is reset, but all actual outputs and simulated inputs/outputs are turned off. - UNSIMULATE I/O power failure recovery is performed, but all the simulated states are reset. This is equivalent to "NOT RECOVER" if the hot start function is disabled, because the output states are not recovered. - RECOVER ALL I/O power failure recovery is performed if the hot start function is enabled. The output and simulated states are recovered to the states that existed immediately before the power is turned off. If the hot start function is disabled, RECOVER ALL is equivalent to "RECOVER SIM", because the output states are not recovered. ⚠ **CAUTION** Even if power failure handling is enabled, the output signal is turned off without being recovered in the following cases: ● When the I/O allocation is changed before the power is turned off. ● When the fuse of an I/O unit blows, or when an I/O unit is turned off. ● When the I/O unit configuration is changed.

Items	Descriptions
COLD START Autoexec program HOT START Autoexec program	Specifies the name of the auto-start program for the hot start or cold start. There are two methods to set the program name. One is the method to select program name in the menu displayed by pressing F4, [CHOICE]. And another is the method to enter the program name directly by pressing [ENTER] key. In order to clear the selection, delete the program name by BACK SPACE key in the method to enter the program name directly. The specified program is executed immediately after the power is turned on. If the auto-start program for the hot start does not end within 15 seconds, it will be aborted. The output does not turn on if I/O Pulse instruction is specified in auto-start program for the hot start because the I/O signal is outputted after the auto-start program execution is finished. ⚠ **CAUTION** The program automatically executed at power on is executed just before the servo power is turned on. Therefore the robot can not be moved by this program. Set the program which sets up the system, and initializes I/O status, and so on. Moreover, the attributes should be set as follows on the program detail screen. Group Mask: [* , * , * , * , * , * , * , *] Ignore pause: [ON]
HOT START done signal	Specifies the digital signal (DO) that is to be output at the hot start. If the hot start is not performed, the digital signal is turned off. This function is disabled if 0 is specified.
Restore selected program	Specifies whether the program selected at turning off the controller is selected after turning on the controller when the cold start is done. When this is set to TRUE, the program selected at the power off is selected after the power on again. When this is set to FALSE, the program is not selected after power on again. This is set to TRUE in standard setting.
Enable UI signals	Selects whether a UI signal is valid or invalid. When this is set to FALSE, the peripheral input signals (UI[1 to 18]) is disabled. See Section 3.3 ”PERIPHERAL I/O”.
START for CONTINUE only	If this item is enabled, the external start signal (START) starts only those programs that have been paused. → See Section 3.3 ”PERIPHERAL I/O”.
CSTOPI for ABORT	If this item is enabled, those programs that are currently running are forcibly terminated immediately upon the input of CSTOPI. → See Section 3.3 ”PERIPHERAL I/O”.
Abort all programs by CSTOPI	Specifies whether all programs are to be forcibly terminated with the CSTOPI signal in a multitasking environment. If this item is set to "TRUE", the CSTOPI input signal terminates all programs forcibly: If this item is set to "FALSE", the CSTOPI input signal causes only the currently selected program to be forcibly terminated. (Default setting)
PROD-START depend on PNSTROBE	If this item is enabled, the PROD_START input is enabled only when the PNSTROBE input is on. By enabling this item, it is possible to prevent a program that should not be started from being started accidentally due to noise or a sequence error when that program is displayed on the teach pendant.
Detect FAULT_RESET signal	Specifies whether the reset signal is detected the instant it rises or falls. When this setting is changed, cycle power of the controller to use the new information. At this time the cold start is done automatically. The falling edge is detected by standard setting.

Items	Descriptions
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, cycle power of the controller to use the new information. At this time, the controller is started up by COLD START. The default setting is FALSE.
WAIT timeout	Specifies the period of time used in the conditional wait instruction "WAIT ..., TIMEOUT LBL[...]". The period of time is 30 second in default setting.
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 FALSE, 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.
Original program name (F1 to F5)	Specifies the words which are displayed as the function 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 [ENTER] 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 ...).
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, circular and circle arc default motion instructions or delete it from them. - Pressing the F4, "ADD" key adds the Wjnt motion option to all the linear, circular and circle arc 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, circular and circle arc 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.

Items	Descriptions
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 FANUC Robot series R-30*i*B/R-30*i*B Plus CONTROLLER MAINTENANCE MANUAL (B-83195EN) or the FANUC Robot series R-30*i*B Mate/R-30*i*B Mate Plus CONTROLLER MAINTENANCE MANUAL (B-83525EN). <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 F4, "TURE". 5) Press the reset button on the teach pendant.
Allow force I/O in AUTO mode	Enables or disables signal setting from TP when AUTO mode is set. By default, setting is enabled. - Yes : Enables signal setting. - No : Disables signal setting.
Allow chg. 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 DO is turned on. When 0 (default) is set, this function is disabled. When the setting has been changed, the power must be turned off then back on.
Signal to set in T1 mode	If the three-mode switch is set to T1 mode, a specified DO is turned on. When 0 (default) is set, this function is disabled. When the setting has been changed, the power must be turned off then back on.
Signal to set in T2 mode	When the three-mode switch is set to T2 mode, a specified DO is turned on. When 0 (default) is set, this function is disabled. When the setting has been changed, the power must be turned off then back on.
Signal to set if E-STOP	When an emergency stop (TP, external emergency stop, operator's panel) is applied, a specified DO is output. When 0 (default) is set, this function is disabled. When the setting has been changed, the power must be turned off then back on.
Simulated status signal (Set if INPUT SIMULATED)	Monitors to see if an input signal set as a simulated one exists and outputs it as an output signal. For this item, set the number of the output signal that will turn on when a digital, group, robot, or analog input signal is set as a simulated one. When the setting has been changed, the power must be turned off then back on.
Simulated status signal (Set if OUTPUT SIMULATED)	Monitors to see if an output signal set as a simulated one exists and outputs it as an output signal. For this item, set the number of the output signal that will turn on when a digital, group, robot, or analog output signal is set as a simulated one. When the setting has been changed, the power must be turned off then back on.
Simulated input wait time (Sim. Input Wait Delay)	Sets the time after which a wait command causes a timeout if the simulated signal skip function is enabled. When the setting is changed, the change is immediately applied.
Setting to be assumed when the simulated signal skip function is enabled (Set if Sim. Skip enabled)	Monitors to see if there is any input signal for which the simulated signal skip function is enabled and outputs the signal as an output signal. For this item, set the number of the output signal that will turn on when the simulated signal skip function is enabled for a digital or robot input signal. When the setting has been changed, the power must be turned off then back on.
Set when prompt displayed	This item provides a digital output signal from the controller that indicates that a prompt box is being displayed on the teach pendant.
Output when WAIT on Input	This item is set for the function that outputs DO when the timeout time is passed during WAIT instruction of program is waiting for specified DI. Move the cursor to this item and press [ENTER] key to display setup screen. About the detail of this function, refer to Subsection "4.8.3 The Output When Wait on Input".

Items	Descriptions
Signal to be output at an override of 100% (Signal if OVERRIDE= 100)	Sets the number of the digital output signal for notifying that the override is set to 100%. For the digital output, on is output if the override is 100% and otherwise, off is output. When the setting has been changed, the power must be turned off then back on.
Hand broken	Enables and disables hand breakage (*HBK) detection. When the system consists of multiple robots, hand breakage detection can be enabled and disabled for each robot. Press [ENTER] key with the cursor positioned on this line to enter the screen for enabling or disabling hand breakage detection for each robot. On this screen, move the cursor to the target item, then press the ENABLE (F4) or DISABLE (F5) key to enable or disable hand breakage detection. For R-30iB or earlier, the default setting is ENABLE. For R-30iB Plus or later, the default setting is DISABLE, but when the *HBK signal is turned on, the setting is automatically set to ENABLE. • When hand breakage detection is enabled, and the *HBK signal is off, alarm "SRVO-006 SERVO Hand broken" is posted. Refer to the subsection “Hand Breakage Recovery” in “FANUC Robot series OPERATOR'S MANUAL (Alarm Code List)” (B-83284EN-1) to release the alarm. When the *HBK signal is off, and this signal is not to be used, disable the hand breakage detection. • When hand breakage detection setting is disabled although a hand is installed, and the *HBK signal is used, "SRVO-302 SERVO Set Hand broken to ENABLE" occurs if the *HBK signal is on. Enable hand breakage detection. For R-30iB Plus or later, "SRVO-307 HBK Close / HBK Enabled" occurs only once instead of SRVO-302, and hand breakage detection is automatically enabled. • If the *HBK signal is turned off when hand breakage detection is disabled, or if you change hand brakeage detection from ENABLE to DISABLE when the *HBK signal is off, "SRVO-300 SERVO Hand broken/HBK disabled" occurs. In this case, this alarm can be released by pressing the RESET key.
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 : At present, this item is not available.
External I/O (ON : Remote)	When "External I/O" is selected in "Remote/Local setup" above, specify an external signal to be used here. Choose from "DI, DO, RI, RO, UI, and UO".
UOP auto assignment	Select the type of UOP auto assignment. Refer to "3.3 PERIPHERAL I/O" for detail. When setting is changed, the message “Clear ALL I/O assignments to apply this?” is displayed. If press F4, "YES", all I/O assignments are cleared. After that, if the controller is cycled power, the UOP is assigned automatically according to the setting of UOP auto assignment.
Multiple-program selection (Multi Program Selection)	This setting is for switching the program selection method between single task mode and multitask mode. If this setting is enabled, multitask mode is selected, and if it is disabled, single task mode is selected. If the R651 standard settings are ordered, the setting is disabled by default, and if the R650 North America only settings are ordered, it is enabled by default. When the setting is changed, the change is immediately applied.
WAIT at Taught Position	When WAIT instruction is executed after motion instruction with CNT, this setting can change wait position of the Robot. If the setting is TRUE, the behavior of Robot is equivalent to the case that WAIT instruction is executed after motion instruction with "BREAK" (Even if BREAK instruction is not taught). (Please refer to “BREAK instruction” in “4.3.5 Additional Motion Instructions” for details.

Items	Descriptions
Brake control ECO mode	Enables or disables Brake control ECO mode. When the ECO mode is enabled (TRUE), the time to brake automatically in AUTO mode is shortened (the default is 5s). When the ECO mode is disabled (FALSE) or T1/T2 mode, the normal brake control time is applied. The default setting is disabled (FALSE). When the ECO mode is enabled, cycle time may be increased compared to the normal brake control because the frequency of brake release is increased. On the axis that has no brake or the brake control is disabled, this setting has no effect. Turn the power off and back on again for the changed settings to take effect.
J7, J8 jog key Setup	This item is the setting to customize the J7, J8 jog keys. Normally, J7, J8 keys are used to jog the extended axes in a group. But, if the setting is changed, it is possible to jog any axis using J7, J8 keys. And, it is possible to disable J7, J8 keys, too. (For detail, refer to "Setting of J7 key and J8 key " in the Subsection “5.2.3 Moving the Robot by Jog Feed”.)
Collection name	Specifies the words which are displayed as the function key at registering a collection program. It is convenient to set the words used many times as the Collection name to this. (For detail, refer to the Subsection “5.8.4 Program Collection Function”.)
Confirmation of TOUCHUP	If the setting is TRUE, the system confirms you when you press F5[TOUCHUP] on the program EDIT screen.
No motion PR operate mode	When this item set to TRUE, position register instructions in non-motion group program can be executed. (About the detail of this function, refer to Subsection "4.5.4 Position Register Instructions in Non-Motion Group Program".)
Use No-display Sub-Program	This feature allows you to make a teach pendant program not display its contents while the program is CALLED as a sub-program or as a macro. When this feature is enabled, a new item appears in the SELECT/DETAIL screen so you can set a program as "NoDisp Sub-Prog".
Export diagnosis data	This item is set for the function that copy Internal data after getting the diagnosis log data. By default, setting is enabled. When Export after saved is set to TRUE, display dialog to copy the data. When save on the dialog, create year month date_ hour minune directory, copy the data. When current device to default is set to TRUE, location to save is current device.
Check Sim. I/O when FWD/BWD	When this item sets to TRUE, the system confirms you when you executes a program with pressing FWD/BWD key on a teach pendant on the condition that there is simulated I/O.
Confirmation for AUTO	Set digital input which is assigned AUTO confirmation signal. This setting is needed to use TP mode select function. Please refer “14.3 TP MODE SELECT FUNCTION” about the function.
Ambient Temperature	This item specifies the current ambient temperature surrounding each robot group. Set the Group Number and the Ambient Temperature in degrees (Celsius). The default group number is 1. The default ambient temperature is 20 degrees Celsius.
Allow I/O chg. in servo-off	Enables or disables output signal setting from TP when servo is not ready. By default, setting is enabled. - TRUE : Enables output signal setting. - FALSE : Disables output signal setting.
Prohibit I/O change anytime	Enables or disables signal setting from TP anytime. By default, setting is FALSE. - FALSE : Enables signal setting. - TRUE : Disables signal setting.
Incremental Jog	When this item sets to TRUE, Incremental Jog function which allows to move the robot to an arbitrary distance/angle at a time by jog feed is enabled. (For detail, please refer to the item “Incremental Jog” in the Subsection “5.2.3 Moving the robot by jog feed”.) Turn the power off and back on again for the changed settings to take effect.

Items	Descriptions
Zero 0% Obverride	When this item sets to TRUE, 0% becomes available as the mininum value of feed rate override. When the feed rate override is 0%, the robots jogging or executing a TP program suspends immediately, and resumes when the feed rate override is changed to non-0%. The feed rate oveeride can be changed to 0% only by the override key.

Procedure 3-33 Setting The System

Step

1 Select [MENU] key. The screen menu will be displayed.
2 Select “6 SYSTEM” in the next page.
3 Press F1, [TYPE]. The screen change menu will be displayed.
4 Select "Config". The system configuration screen will be displayed.

System configuration screen

```
System/Config
                                           1/49
      1 Use HOT START                      TRUE
      2 I/O power fail recovery:     RECOVER ALL
      3 COLD START Autoexec program:
       [****************************************]
      4 HOT START Autoexec program:
       [****************************************]
      5 HOT START done signal:          DO[    0]
      6 Restore selected program:          TRUE
      7 Enable UI signals:                 TRUE
      8 START for CONTINUE only:          FALSE
      9 CSTOPI for ABORT:                 FALSE
     10 Abort all programs by CSTOPI:     FALSE
     11 PROD_START depend on PNSTROBE:    FALSE
     12 Detect FAULT_RESET signal:         FALL
     13 Use PPABN signal:            <*GROUPS*>
     14 WAIT timeout:                 30.00 sec
     15 RECEIVE timeout:              30.00 sec
     16 Return to top of program:          TRUE
     17 Original program name(F1):     [RSR     ]
     18 Original program name(F2):     [PNS     ]
     19 Original program name(F3):     [STYLE   ]
     20 Original program name(F4):     [JOB     ]
     21 Original program name(F5):     [TEST    ]
     22 Default logical command:     <**DETAIL**>
     23 Maximum of ACC instruction:         150
     24 Minimum of ACC instruction:           0
     25 WJNT for default motion:           ****
     26 Auto display of alarm menu:       FALSE
     27 Force Message:                   ENABLE
     28 Reset CHAIN FAILURE detection:    FALSE
     29 Allow Force I/O in AUTO mode:      TRUE
     30 Allow chg. ovrd. in AUTO mode:     TRUE
     31 Signal to set in AUTO mode      DO[    0]
     32 Signal to set in T1 mode        DO[    0]
     33 Signal to set in T2 mode        DO[    0]
     34 Signal to set if E-STOP         DO[    0]
     35 Set if INPUT SIMULATED          DO[    0]
     36 Set if OUTPUT SIMULATED         DO[    0]
     37 Sim. Input Wait Delay:          0.00sec
     38 Set if SIm. Skip Enabled:       DO[    0]
     39 Set when prompt displayed:      DO[    0]
```

```
   40 Output when WAITing on Input  <**DETAIL**>
   41 Signal if OVERRIDE = 100        DO[    0]
   42 Hand broken :                   <*GROUPS*>
   43 Remote/Local setup                   LOCAL
   44   External I/O(ON:Remote):      DI[    0]
   45 UOP auto assignment:                  Full
   46 Multi Program Selection:             FALSE
   47 WAIT at Taught Position:             FALSE
   48 Brake control ECO mode:              FALSE
   49 J7, J8 jog key Setup:         <**DETAIL**>
   50 Collection program name(F1):    [STYLE   ]
   51 Collection program name(F2):    [COL     ]
   52 Collection program name(F3):    [********]
   53 Collection program name(F4):    [********]
   54 Collection program name(F5):    [********]
   55 Confirmation of TOUCHUP:             FALSE
   56 No motion PR operate mode:           FALSE
   57 Use No-display Sub-Program:          FALSE
   58 Export diagnosis data:          <*DETAIL*>
   59 Check Sim. I/O when FWD/BWD:         FALSE
   60 Confirmation for AUTO:          DI[    0]
   61 Ambient Temperature:            <*DETAIL*>
   62 Allow I/O chg. in servo-off:          TRUE
   63 Prohibit I/O change anytime:         FALSE
   64 Incremental Jog:                <*DETAIL*>
   65 Zero 0% Override:                    FALSE

 [ TYPE ]                TRUE     FALSE
```

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 [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 [ENTER] key. Then each setting screen is displayed. Press PREV key to get out of these screens.

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.)

```
System/Config
                                        12/47

   12 Detect FAULT_RESET signal:  FALL
   13 Use PPABN signal:  <*GROUP*>
   14 WAIT timeout:30.00 sec

   Please power on again

 [ TYPE ]                RISE      FALL
```

3.16 SETTING THE GENERAL ITEMS

[6 SETUP General] has the following items.

- Break on hold
- Current language
- Ignore Offset command
- Ignore Tool_offset
- Enable VOFFSET

Table 3.16 Setting the general items

Items	Descriptions
Break on hold	Specifies whether to issue an alarm and turn off the servo power 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. **NOTE** This function cannot be used together with interference prevention area function. If you use interference prevention area function, please disable this function. If you use this function, please disable interference prevention area function.
Current language	In case that the Option Dictionary option is loaded, it is possible to change the language by pressing F4, [SELECT]. When the language is changed, cycling power of the controller is required. In case that this item is displayed as “DEFAULT”, because the Option Dictionary option is not loaded, the language cannot be changed.
Ignore Offset command	Specifies whether to ignore the offset command (→Subsection 4.3.5 Additional Motion Instructions”). • If the function is "DISABLED", the robot moves to the position for which the offset command has been executed (standard setting). • If the function is "ENABLED", the robot moves to the taught position (for which the offset command has not been executed). See Subsection 4.3.5 ”Additional Motion Instructions” for details of the Offset command.
Ignore Tool_offset	Specifies whether to ignore the tool offset command (→Subsection 4.3.5 ”Additional Motion Instructions”). • If the function is "DISABLED", the robot moves to the position for which the tool offset command has been executed (standard setting). • If the function is "ENABLED", the robot moves to the taught position (for which the tool offset command has not been executed). See Subsection 4.3.5 ”Additional Motion Instructions” for details of the Tool offset command.
Enable VOFFSET	Specifies whether to enable the vision offset command. • If the setting is "ENABLED", the robot moves to the position to which the vision offset is applied (Standard setting). • If the setting is "DISABLED", the robot moves to the originally taught position.

Procedure 3-34 Setting the general items

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select “6 SYSTEM”.
3 Press F1, [TYPE]. The screen change menu will be displayed.
4 Select "General".

General item setting screen

```
SETUP General
                                                 1/5
       1 Brake on hold:                DISABLED
       2 Current language:              DEFAULT
       3 Ignore Offset command:        DISABLED
       4 Ignore Tool_offset:           DISABLED
       5 Enable VOFFSET:               DISABLED

   [ TYPE ]                    ENABLED  DISALBED
```

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, cycle power of the controller. The setting of the other functions is made effective immediately when they are reset.

3.17 PAYLOAD SETTING

Overview

Payload setting is a setting of the payload which is mounted on the robot. It consists of payload data such as weight, center of gravity, etc. Setting appropriate payload data can bring the following effects:

- Improvement in motion performance (such as reduction of vibration and cycle time)
- More effective performance of functions related to dynamics (such as collision detection and gravity compensation).

Wrong payload data can cause vibration, mis-detection of collision, and so on. For effective use of the robot, set appropriate data of the load such as a hand, a workpiece, and an armload (equipment mounted on the robot arm, not on the face plate).

You can set the payload data using “Motion Performance screens". In these screens, you can set up 10 schedules of payload setting. If you set up two or more payload schedules in advance, you can achieve appropriate payload setting only by switching the schedule number according to changing of actual payload. You can also change the schedule number using program instruction which you can place anywhere in TP program. (Refer to the Subsection 4.10 PAYLOAD INSTRUCTION.)

As an option function, “Payload Identification” is available. This function enables the robot to calculate load information automatically.

Motion Performance Screens (Payload setting screen)

You can set the payload data using “Motion Performance screens". Motion Performance screens consist of list screen, payload setting screen and armload setting screen.

Table 3.17 Motion Performance screen

Screen name	Description
MOTION PERFORMANCE (List screen)	A screen to display the list of payload schedules (No. 1 ~ No. 10). You can also check or switch the active schedule number in this screen.
MOTION / PAYLOAD SET	A screen for detailed payload data for each schedule. You can display or modify the values of payload weight, gravity center position and inertia in this screen. This screen is provided for each individual payload setting number.
MOTION / ARMLOAD SET	A screen for armload (equipments mounted on the robot arm, not on the face plate). You can set the weight of equipments mounted on such as J2 base and J3 arm.

Procedure 3-35 Setting payload data

The following description is a procedure to show Motion Performance screen, to input payload data and to activate the payload schedule. You can also modify the setting values later.

> **NOTE**
> Depending on the combination of robot model and option, the payload data may be set in advance.

Step

1 Press MENU key to display the screen menu.
2 Press “0 NEXT”, then select “6 SYSTEM”.
3 Press F1 [TYPE] to display the screen switch menu.
4 Select “Motion”. The list screen (“MOTION PERFORMANCE” screen) appears. (If a screen other than the list screen appears, press PREV key several times until the list screen appears.) For a multi-group system, if you want to go to the list screen of the other group, press F2, "GROUP" then enter the group number you like.

```
MOTION PERFORMANCE
        Group1                         1/10
     No.    PAYLOAD[kg]          Comment
      1          50.00     [                 ]
      2          50.00     [                 ]
      3          50.00     [                 ]
      4          50.00     [                 ]
      5          50.00     [                 ]
      6          50.00     [                 ]
      7          50.00     [                 ]
      8          50.00     [                 ]
      9          50.00     [                 ]
     10          50.00     [                 ]
    Active PAYLOAD number = 0

   [ TYPE ]   GROUP   DETAIL   ARMLOAD   SETIND   >
```

NOTE

Up to 10 of payload data can be set as standard. The number of payload data can be increased up to 256 by the following way. (up to 32 in R-30*i*B and R-30*i*B Mate)

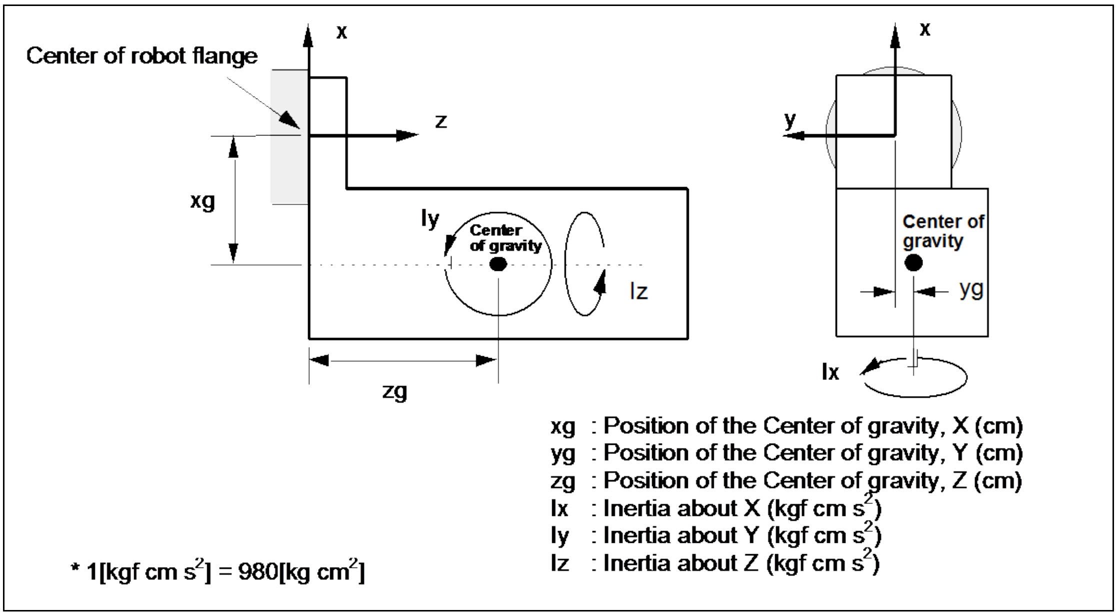
a. Turn on the controller with [PREV] and [NEXT] key pressed.
b. Select “3 Controlled start”.
c. Press the [MENU] key.
d. Select “4 Variables”
e. Set the number of payload data to the system variable $PLST_SCHNUM.
f. Restart (power off/on) the controller and "Control start" again.
g. Press the [FCTN] key.
h. Select “1 START (COLD)”.

5 Move the cursor to the schedule number you want to set up, and press F3, "DETAIL" to display PAYLOAD SET screen of the selected schedule number.

```
MOTION/PAYLOAD SET
                                          1/8
         Group 1
    1 Schedule No[    1]:[****************]
    2 PAYLOAD              [kg]         50.00
    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 Enter the weight, gravity center position of the load, and inertia about its gravity center.
If required, you can enter the comment. The entered comment is displayed in list screen.

The X, Y, and Z directions displayed in the PAYLOAD SET screen correspond to those in the default tool coordinate system (“default” means no Tool frame is set).



When you change the value, a confirmation message "Path and Cycle time will change. Set it?" appears. Press F4, "YES" or F5, "NO".
Then, you may see a message "Load is OVER spec! Accept?". This message indicates the load exceeds the capacity of the robot. Reconfigure your system so that the load does not exceed the capacity.
In another case, you may see a message "Load is close to capacity! Accept?". This message indicates the load is close to capacity although it does not exceed the capacity.

> ⚠ **CAUTION**
> Don't install a load which exceeds the capacity of the robot because overload can cause a life loss of the reducer.

7 If you want to go to the screen of the other schedule number, press F3, "NUMBER" then enter the schedule number you like. For a multi-group system, if you want to go to the screen of the other group, press F2, "GROUP" then enter the group number you like.
8 Activate payload schedule you are going to use. Press PREV to go back to the list screen, press F5, "SETIND", and enter the index number of payload schedule you are going to use.

> **NOTE**
> The initial schedule number is 0. In order to activate the payload setting you entered, you need to set the schedule number to 1-10. If the schedule number is 0, the system uses the default payload value which is shown in the screen before you change the value.

9 Press F4, "ARMLOAD" on the list screen to go to the ARMLOAD SET screen.

```
MOTION/ARMLOAD SET
                                             1/2
        Group 1
      1 ARM LOAD AXIS #1   [kg]             0.00
      2 ARM LOAD AXIS #3   [kg]             0.00




   [ TYPE ]   GROUP             DEFAULT    HELP
```

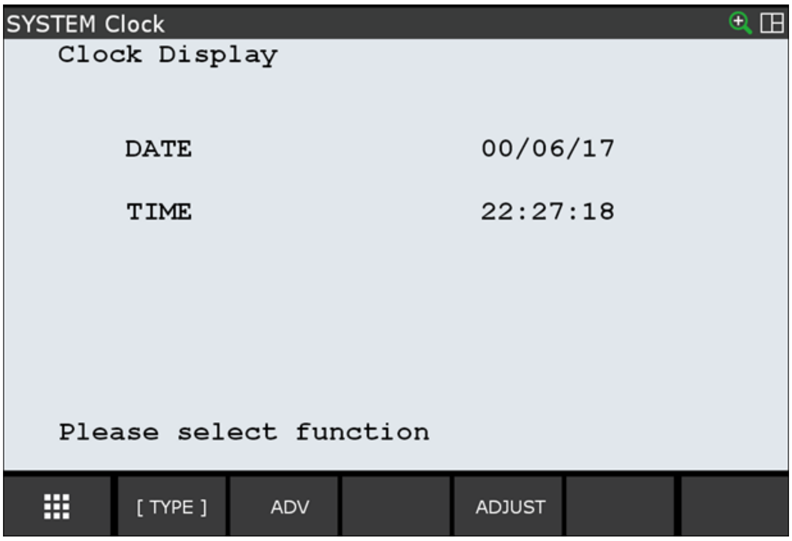
10 Enter the weight of the equipment in each item. These item names vary depending on the robot model. In the example image above, ARM LOAD AXIS #1 is the mass of the equipment mounted on the J2 base and ARM LOAD AXIS #3 is the mass of the equipment mounted on the J3 arm. Please refer to the MECHANICAL UNIT operator's manual for the item names and mounting positions of each robot model.
When you change the value, a confirmation message "Path and Cycle time will change. Set it?" is displayed. Press F4, "YES" or F5, "NO".
If you change the armload value, cycle power of the controller.

> **NOTE**
> For armload, there is one schedule available. The input value is always valid regardless of the active payload schedule number.

3.18 CLOCK

SYSTEM Clock screen shows the date and time of the calendar clock on the teach pendant.

> **NOTE**
> When the main board is replaced, please set the clock.



Procedure 3-36 Displaying the clock

Step

1. Press the [MENU] key to display the screen menu.
2. Press “0 -- NEXT --”.
3. Select “6 SYSTEM”.
4. Press the F1, [TYPE] to display the screen change menu.
5. Select “Clock” to display the clock display screen.

Procedure 3-37 Adjusting the clock

Step

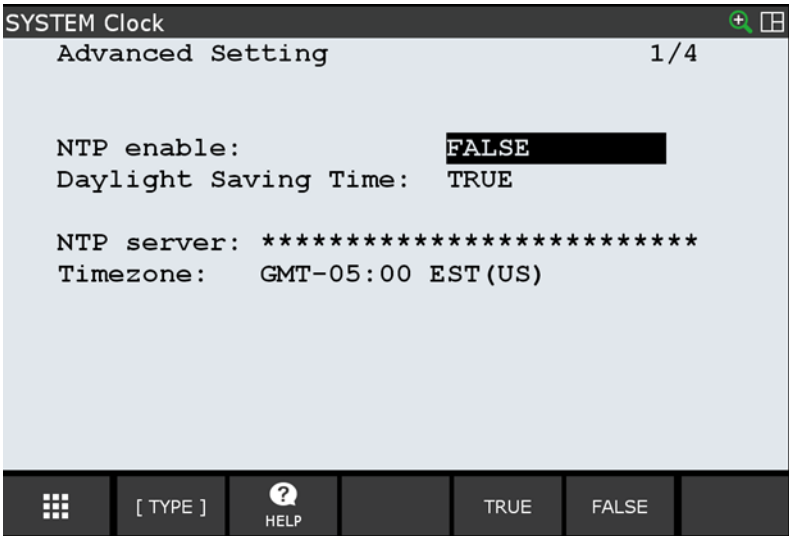
1. Display the clock display screen according to the Procedure 3-36.
2. Press the F4, ADJUST.
3. Enter the date and time
4. Press the F4, FINISH.

> **NOTE**
> When you adjust the clock, the WARN alarm “SYST-136 System Time was reset” will be displayed.

By using SNTP, the robot (SNTP client) gets the current date and time from a central NTP/SNTP server.

> **NOTE**
> SNTP can be used with R-30iB Plus or later controllers by default, but with R-30iB or earlier controllers, Internet Connectivity and Connection option is required.

Setting up SNTP can be performed on the advanced setting screen.
Press the F2, ADV on the clock display screen to display the advanced setting screen.



The local system clock is updated by the remote Network Time Protocol (NTP) server. It requires a user to input following fields.

Item	Description
NTP enable	Set it TRUE after all other fields are filled in. Press the F4, TRUE or F5, FALSE to switch between TRUE and FALSE.
Daylight Saving Time (DST)	The local time is adjusted to DST while DST is in effect. Press the F4, TRUE or F5, FALSE to switch between TRUE and FALSE.
NTP server	Please contact your network administrator to get NTP server address. If DHCP is enabled and configured to return NTP server address, this field can be set by DHCP.
Timezone	Press the F4, [CHOISE] to choose time zone from sub-menu.

Explanation of each area is displayed by pressing the F2, HELP. To display the previous screen, press the [PREV] key.

NOTE
Refer to the “Ethernet Function OPERATOR’S MANUAL” (B-82974EN) for more information on SNTP.

3.19 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 controlled start is required.)
- The system is in remote control state.
- The speed override is not changed while the safety fence is open.

Other items are set up on the system variable screen, [6 SYSTEM Variables]. To specify system variables, see the appropriate appendix (→ Appendix C, SYSTEM VARIABLES).

4 PROGRAM STRUCTURE

This chapter describes the program structure and program instructions.

Contents of this chapter

A robot application program consists of commands described by the user for the robot to perform operations and other incidental information.
In addition to program information that describes how the robot should perform operations, a program contains program detail information that defines program attributes.

```
Program detail
                                                  1/7
      Creation Date:          16-Jan-1994
      Modification Date:      08-Mar-1994
      Copy Source:

      Positions: FALSE  Size:      312 Byte

      Program name:
       1 SAMPLE3
       2  Sub Type:      [None             ]
       3  Comment:       [SAMPLE PROGRAM 3 ]
       4  Group Mask:    [1,*,*,*,*,*,*,*  ]
       5  Write protect: [OFF              ]
       6  Ignore pause:  [OFF              ]
       7  Stack size:    [     500         ]

       END     PREV     NEXT
```

Fig. 4 (a) Program information screen

Program detail information consists of the following information items:

- Attribute-related information items such as a creation date, modification date, a copy source file name, presence/absence of position data, and program data size.
- Information items related to an execution environment such as a program name, subtype, comment, group mask, write protection, interruption disable and stack size.

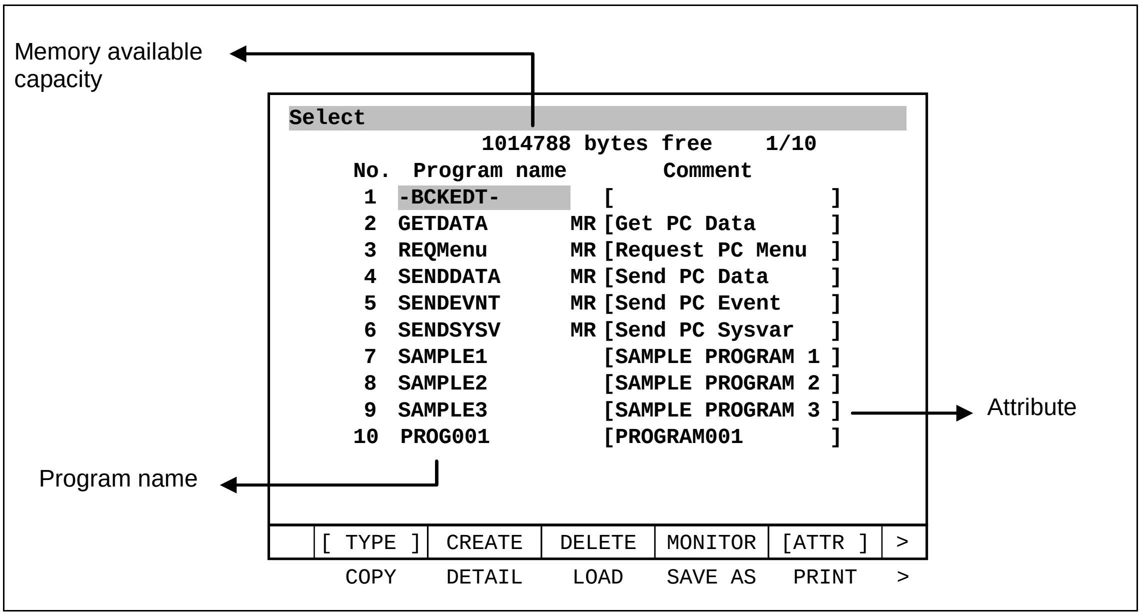


Fig. 4 (b) Program selection screen

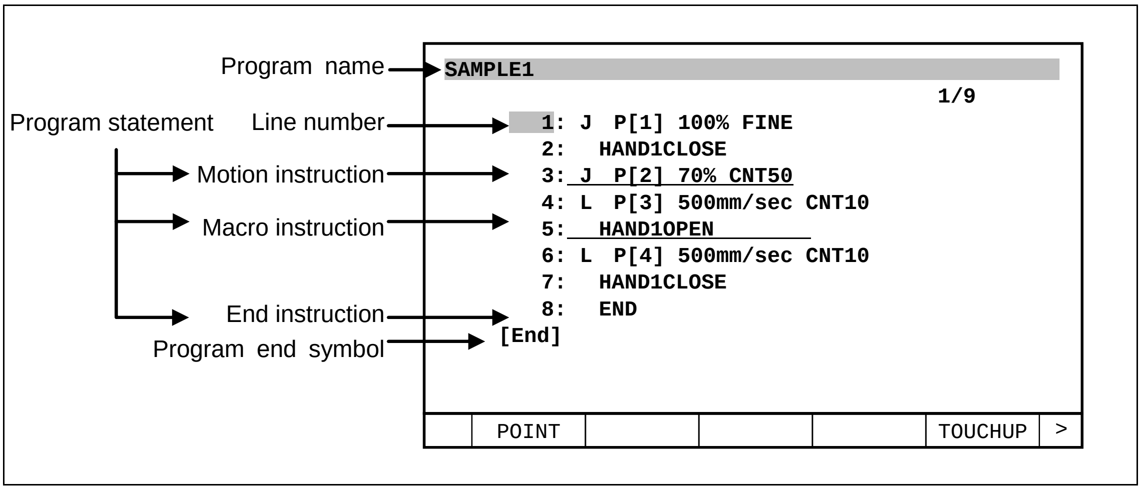


Fig. 4 (c) Program edit screen

A program consists of the following information:

- Line number assigned to each program command
- Motion instructions specifying how and where the robot is to move
- Program instructions including the following:
  - Palletizing instructions for performing the palletizing function
  - Instructions for storing numerical data in registers (register instructions)
  - Instructions for storing robot position data in position registers (position register instructions)
  - I/O instructions to output and input signals to and from peripheral devices
  - Branch instructions for changing the flow of program control when a defined condition is satisfied (IF, JMP/LBL, CALL/END)
  - Wait instructions for suspending program execution
  - Skip condition instruction for operating the robot until a signal is received. If the signal is not received, a branch to a specified command occurs. If the signal is received, the next command is executed, canceling the operation.
  - Program comments
  - Other instructions
- Program end symbol indicating that the program contains no more instructions

Program detail information is set on the program information screen. (See Subsection 5.3.1 and 5.5.) A program is registered on the program registration screen. (See Subsection 5.3.1.) A program is created and changed on the program edit screen. (See Sections 5.3 and 5.4.)

4.1 PROGRAM DETAIL INFORMATION

Program detail information names a program and defines the attributes of the program. Program detail information consists of the following items:

- Attribute-related information items such as a creation date, modification date, a copy source file name, presence/absence of position data, and program data size.
- Information items related to an execution environment such as a program name, subtype, comment, group mask, write protection, interruption disable and stack size. (There is an item “collection” if system software version is 7DC3 series or later)

```
Program detail
                                          1/7
      Creation Date:          16-Jan-1994
      Modification Date:      08-Mar-1994
      Copy Source:

      Positions: FALSE  Size:      312 Bytes

       Program name:
       1 SAMPLE3
       2  Sub Type:     [None             ]
       3  Comment:      [SAMPLE PROGRAM 3 ]
       4  Group Mask:   [1,*,*,*,*,*,*,*  ]
       5  Write protect:[OFF              ]
       6  Ignore pause: [OFF              ]
       7  Stack size:   [     500         ]
       8  Collection:   [LINE1            ]

   END  |  PREV  |  NEXT  |        |
```

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 setting, can be displayed on the selection screen by pressing F5, [ATTR] and selecting the item from a pull up menu.

4.1.1 Program Name

A program name is specified to identify a program stored in the memory of the controller. A single controller cannot contain two or more programs with the same program names.

Length

A program name must consist of one to 36 characters. A unique name must be assigned to each program.

> ⚠**CAUTION**
> If 36 characters cannot be displayed, the last character of program name is “>” example for “ABCDEF>” in this situation, complete program name is displayed in prompt line.

Usable characters

Character: Alphabetic characters, Number: 0 to 9. No program name can start with a number. Symbol: Underscore (_) only. The at mark (@) and asterisk (*) cannot be used.

Unusable program name

The following program names are not usable.
CON, PRN, AUX, NUL
COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9
LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9

Informative name

A program should be named so that purpose or function of the program can be known from its name.
When a program for spot workpiece A is to be named, for example, “SPOT_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.
- **Condition:**
  Specify this when creating a condition program with the state monitoring function.
- **Collection (CO):**

Programs having this subtype display only programs being registered to the collection. Those programs having this subtype don't function as robot application program.(→refer to "5.8.4 Program Collection Function")(*The subtype collection exists on system software version 7DC3 or later)

4.1.4 Motion Group

A motion group sets up an operation group of a program. An operation group represents a group of different axes (motors) used for independent robots, positioning tables, and other jigs.

> **NOTE**
> A motion group must be set before the program is executed.

The robot controller can control multiple axes, divided into multiple operation groups (multiple motion 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 can 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).

> **NOTE**
> When TP program which has no motion group calls TP program which has motion group, lock (when calling) and unlock (when finishing) of motion control are done. In case that this process is done repeatedly, it takes long time for this process and next motion cannot start immediately. To avoid this issue, please set same motion group for both caller program and called program.

Example

MAIN.TP (<*,*,*,*,*,*,*,*>)

```
R[1] = 0;
LBL[1];
CALL SUB1;
R[1] = R[1]+1;
IF R[1] < 1000, JMP LBL[1];
CALL SUB2;
```

SUB1.TP (<1, *,*,*,*,*,*,* >)
TP Program which doesn't have motion sentence

When calling, motion control is locked.
When returning, motion control is unlocked.

SUB2.TP<1, *,*,*,*,*,*,* >
TP Program which has motion sentence

If above lock and unlock of motion control is done repeatedly, it takes long time until completion of process and next motion cannot start immediately.

<Countermeasure>
Please set group mask of MAIN.TP <1, *,*,*,*,*,*,*> or set group mask of SUB1.TP <*,*,*,*,*,*,*,*>.

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 Ignore Pause

Ignore pause allows you to specify whether the program will continue to run even when an error occurs, emergency stop, or hold.
When these signals are to be ignored, set ignore pause to ON.
Ignore pause is effective only in programs that don't have motion groups.
Pause instruction (refer to 4.14.1) in the program can pause the program that has ignore pause attribute.
You can abort the program that has ignore pause attribute.
When an error with ABORT severity occurs, program execution will abort regardless of ignore pause setting.

> ⚠ **WARNING**
> When ignore pause is set to ON, a program being executed cannot be paused by pressing the emergency stop or hold button on the teach pendant or operator's panel.

4.1.7 Stack Size

Stack size specifies the memory size to use when the program call is executed.
If the following stack overflow alarm occurs at execution of program call instruction, the program needs more stack to execute program call instruction. Please increase the stack size in the program detail screen.
At first, set 500 to the stack size. If the alarm still occurs, then set 800, and 1000 to the stack size.

INTP-222 Call program failed.
INTP-302 Stack overflow.

Procedure 4-1 Program Detail Information

Condition

- The teach pendant must be enabled.

Step

1. Press [MENU] key. The screen menu will be displayed.
2. Select "SELECT". The program selection screen will be displayed.
   The program selection screen can be displayed by pressing the "SELECT" key without using above steps.

```
Select
                  1014788 bytes free     1/10
      No.  Program name         COMMENT
       1  -BCKEDT-         [                 ]
       2  GETDATA      MR [Get PC Data       ]
       3  REQMENU      MR [Request PC Menu   ]
       4  SENDDATA     MR [Send PC Data      ]
       5  SENDEVNT     MR [Send PC Event     ]
       6  SENDSYSV     MR [Send PC Sysvar    ]
       7  SAMPLE1         [SAMPLE PROGRAM 1 ]
       8  SAMPLE2         [SAMPLE PROGRAM 2 ]
       9  SAMPLE3         [SAMPLE PROGRAM 3 ]
      10  PROG001         [PROGRAM001       ]

 [ TYPE ]  CREATE   DELETE   MONITOR  [ATTR ]  >
   COPY    DETAIL    LOAD    SAVE AS   PRINT   >
```

- Switching the screen using sub type

3 To specify the program to be displayed for the sub type, press F1, [TYPE] and select the sub type of the program you want to display.

- "All" : All the programs are displayed.
- "TP Program" : All the programs except the macro are displayed.
- "Macro" : Only macro programs are displayed.
- "Cond" : Only condition programs are displayed.
- "Collections" : Only collections are displayed.

- Switching the display using the attribute

4 To specify the program attribute to be displayed, press F5, [ATTR] and select the attribute type of the program you want to display.

- "Comment" : The comment is displayed.
- "Protection" : The setting of the write protection is displayed.
- "Last Modified" : The latest date of the modification is displayed.
- "Size" : The number of the line and the program size are displayed.
- "Copy Source" : The name of the copy source program is displayed.
- "Name Only" : Only the name of program is displayed.

- Program Detail Screen

5 Press NEXT, ">" and press F2, "DETAIL" in the next page. The program detail screen is displayed.

```
Program detail
                                                1/7
      Creation Date:          16-Jan-1994
      Modification Date:      08-Mar-1994
      Copy Source:

      Positions: FALSE  Size:      312 Bytes

       Program name:
       1 SAMPLE3
       2  Sub Type:      [None             ]
       3  Comment:       [SAMPLE PROGRAM 3 ]
       4  Group Mask:    [1,*,*,*,*,*,*,*  ]
       5  Write protect: [OFF              ]
       6  Ignore pause:  [OFF              ]
       7  Stack size:    [     500         ]

    END      PREV     NEXT
```

6 When you finish setting the program header information, press F1, "END".

4.1.8 Collection

- Collection Editor menu is displayed to register this program's child programs when the program has subtype collection.
- Collection list menu is displayed to register itself to a child of collection program when the program does not have subtype collection.
  (→refer to "5.8.4 Program Collection Function").

> **NOTE**
> There is an item "collection" on the program detail screen if system software version is 7DC3 or later.

4.2 LINE NUMBER, PROGRAM END SYMBOL, AND ARGUMENT

Line number

A line number is automatically inserted in front of an instruction when it is added to a program. When an instruction is deleted, or an instruction is moved to another location, the lines of the program are renumbered in ascending order; that is, the first line is numbered as 1, the second line is numbered as 2, and so forth.
When a program is to be modified, the cursor can be used to specify a line or a range of lines for movement or deletion by line number.
The user can make the cursor move to a desired line number by specifying a line number (with the ITEM key).

Program end symbol

The program end symbol ([End]) is automatically displayed on the line after the last instruction of a program. Whenever a new instruction is added, the program end symbol moves downward on the screen. As a result, it is always displayed on the last line.
When the execution of a program reaches the program end symbol after the last instruction in the program is executed, the program execution automatically returns to the first line of the program for termination. However, when the setting of Return to top of program is FALSE, the cursor stays at the last line of the program after program execution is completed. (See Section 3.15 SYSTEM CONFIG MENU.)

A description of the program instructions required to create and change a program follows. (For how to create a program, see Section 5.3. For how to change a program, see Section 5.4.)

Argument i

Argument i is an index used in teaching control instructions (program instructions other than motion instruction). Some arguments are specified directly; others are specified indirectly. In direct specification, an integer from 1 to 32766 is usually specified. The range of values used depends on the type of instruction. In indirect specification, the register number is specified.

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]])

R[1] Register screen

```
DATA Register
                                         1/200
  R[   1:                  ]=11
  R[   2:                  ]=0
```

Fig. 4.2 Format of argument i

Procedure 4-2 Program Edit Screen

Condition

- The teach pendant must be enabled.

Step

1. Display the program selection screen.
2. Move the cursor to the program you want to edit and press the [ENTER] key. The program edit screen will be displayed.

```
SAMPLE1
                                   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  >
```

- Moving the cursor

3. To move the cursor, use the arrow keys such as up, down, right, and left. To move quickly through the information, press and hold [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.

```
SAMPLE1
                                   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: P  P[5] 100% FINE
   [End]

  POINT                    TOUCHUP  >
```

- Entering the numerical value

5. To enter the numerical value, move the cursor to the argument and press the numerical value keys. When you are finished, press the [ENTER] key.

```
SAMPLE1
                                          10/11

      10:   DO[1]=...
     [End]


     Enter value
       DIRECT   INDIRECT  [CHOICE]  [LIST]
```

6 To use the indirect addressing with the register, press F3, "INDIRECT".

```
SAMPLE1
                                          10/11

      10:   DO[R[1]]=...
     [End]


     Enter value
       DIRECT   INDIRECT  [CHOICE]  [LIST]
```

4.3 MOTION INSTRUCTIONS

A motion instruction moves a robot tool to a specified point within the operating area at a specified feed rate and in a specified traveling mode. The items listed below must be specified in a motion instruction. The format of a motion instruction is shown in Fig. 4.3.

- Motion format: Specifies how to control the path of motion to a specified position.
- Position data: Teaches a position to which the robot is to move.
- Feed rate: Specifies the feed rate of the robot.
- Positioning path: Specifies whether to position the robot at a specified point.
- Additional motion instruction: Specifies the execution of an additional instruction while the robot is in motion.

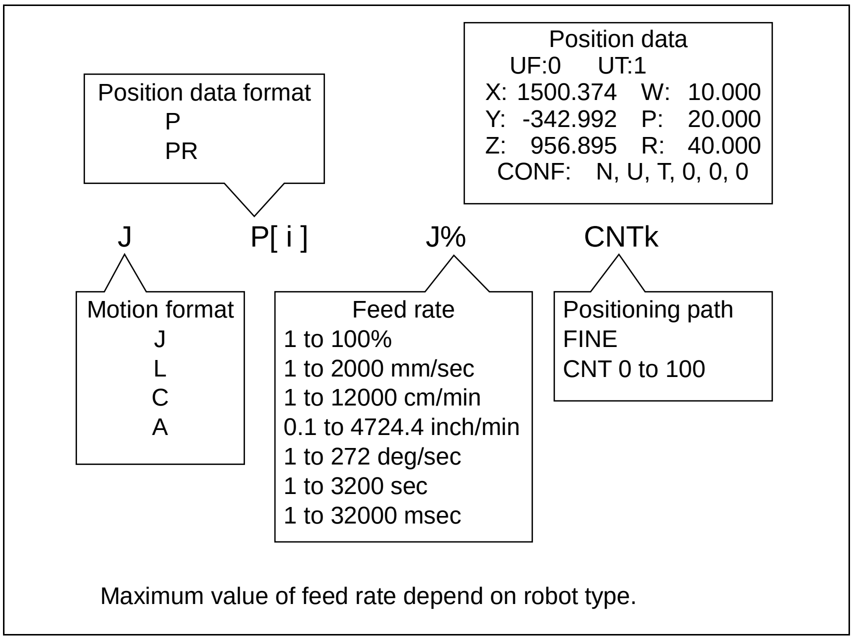


Fig. 4.3 Motion instructions

In teaching a motion instruction, a standard motion instruction is selected using either the F1 or F5 function key. (For modifying a standard motion instruction, see Subsection 5.3.2. For teaching a motion instruction, see Subsection 5.3.3. For changing a motion instruction, see Subsection 5.4.2.)

	POINT				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. Four options are available: joint motion, which does not exercise path/attitude control and linear motion, circular motion and circle arc motion, which exercise path/attitude control.

- Joint motion (J)
- Linear motion (including the rotation motion) (L)
- Circular motion (C)
- Circle arc motion (A)

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 feed rate, 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 feed rate is specified as the feed rate of joint motion. The feed rate of joint motion can be also specified in sec or msec. The attitude of a tool being moved is not controlled.

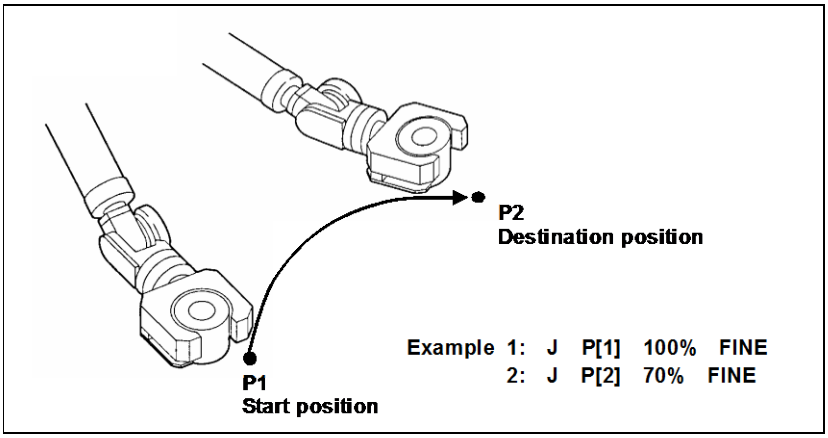


Fig. 4.3.1 (a) Joint motion

Linear motion L

The linear motion mode controls the path of tool center point (TCP) motion from a start point to an end point; the tool center point moves linearly. The motion format is specified to teach an end point. For linear feed rate specification, a desired option must be chosen from mm/sec, cm/min, inch/min, sec and msec. The attitude of a tool being moved is controlled by distinguishing the attitude at a start point from the attitude at a target point.

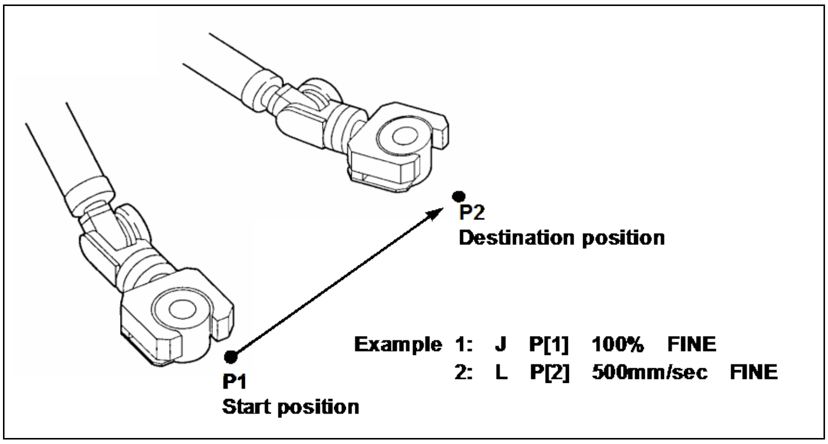


Fig. 4.3.1 (b) Linear motion

Rotary operation is a method of travel in which the tool is rotated about the tool center point from the start position to the end position by using linear operation. The orientation of the tool during travel is controlled by dividing the orientation at the start position and that at the destination position. The feed rate is specified in deg/sec. The focus is controlled linearly (if the tool endpoint moves).

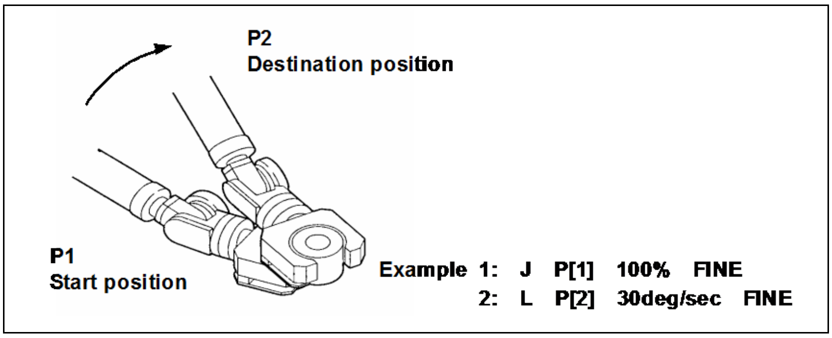


Fig. 4.3.1 (c) Rotation motion

Circular motion C

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 feed rate specification, a desired option must be chosen from mm/sec, cm/min, inch/min, sec, and msec.
The attitude of a tool being moved is controlled by distinguishing the attitude at a start point from the attitude at a target point.

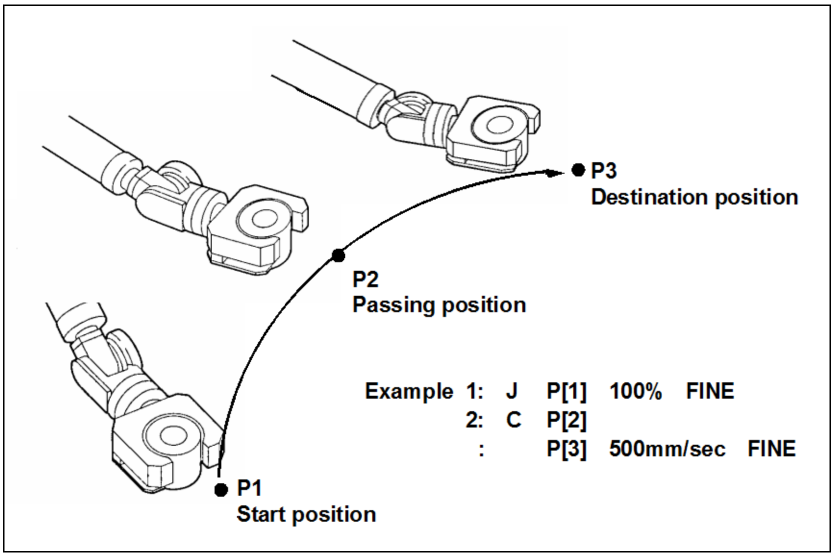


Fig. 4.3.1 (d) Circular motion

Circle arc motion A

The circular motion instruction requires to teach the passing point and end point in one line. Meanwhile, the circle arc motion requires to teach only one position in one line. The circle arc motion is executed connecting the circular arc which is generated from the continued three circle arc motion instructions. For detail, refer to "9.19 CIRCLE ARC MOTION INSTRUCTION".

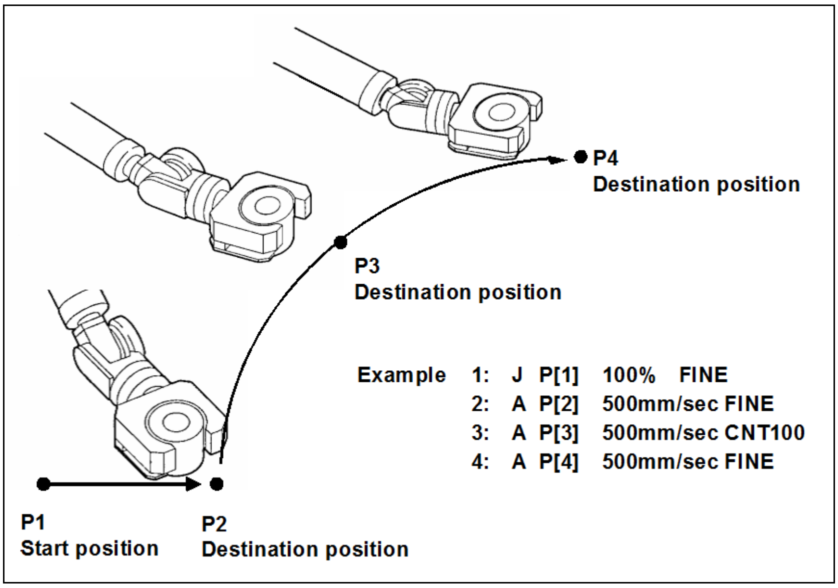


Fig. 4.3.1 (e) Circle arc motion

Motion of 5-axis robot

Since 5-axis robot cannot move under a full control of its Tool attitude, there are many attitudes which the robot cannot reach. For this reason, the motion of 5-axis robot has some unique characteristics.

1 During Linear motion or Circular motion or Circle Arc motion, 5-axis robot can exactly control its Tool attitude ONLY IF the flange surface faces in the vertical direction. If not, 5-axis robot moves in an alternative reachable Tool attitude. This means the robot does not keep the same Tool attitude during the motion. On the other hand, 5-axis robot always keeps its TCP position exactly.

2 5-axis robot has a special definition of Configuration; Joint placement of wrist axis. (See Subsection 4.3.2 for Configuration.)

(1) M-410*i*B/140H, M-710*i*C/50H:
If the flange surface downward, the configuration is NOFLIP. If not, the configuration is FLIP.

(2) F-100*i*A 5-axis:
The definition of FLIP/NOFLIP is special. (See Fig. 4.3.1(f).)
(3) Other 5-axis robot:
If the flange surface faces inward, the configuration is NOFLIP. If not the Configuration is FLIP.

If Configuration of a start point is different from that of a destination point, 5-axis robot cannot perform any Cartesian motion (Linear motion, Circular motion and Circle Arc motion). In this case, use a Joint motion instead or add Wrist joint motion instruction to the Cartesian motion. (see Subsection 4.3.5 for Wrist joint motion instruction.)

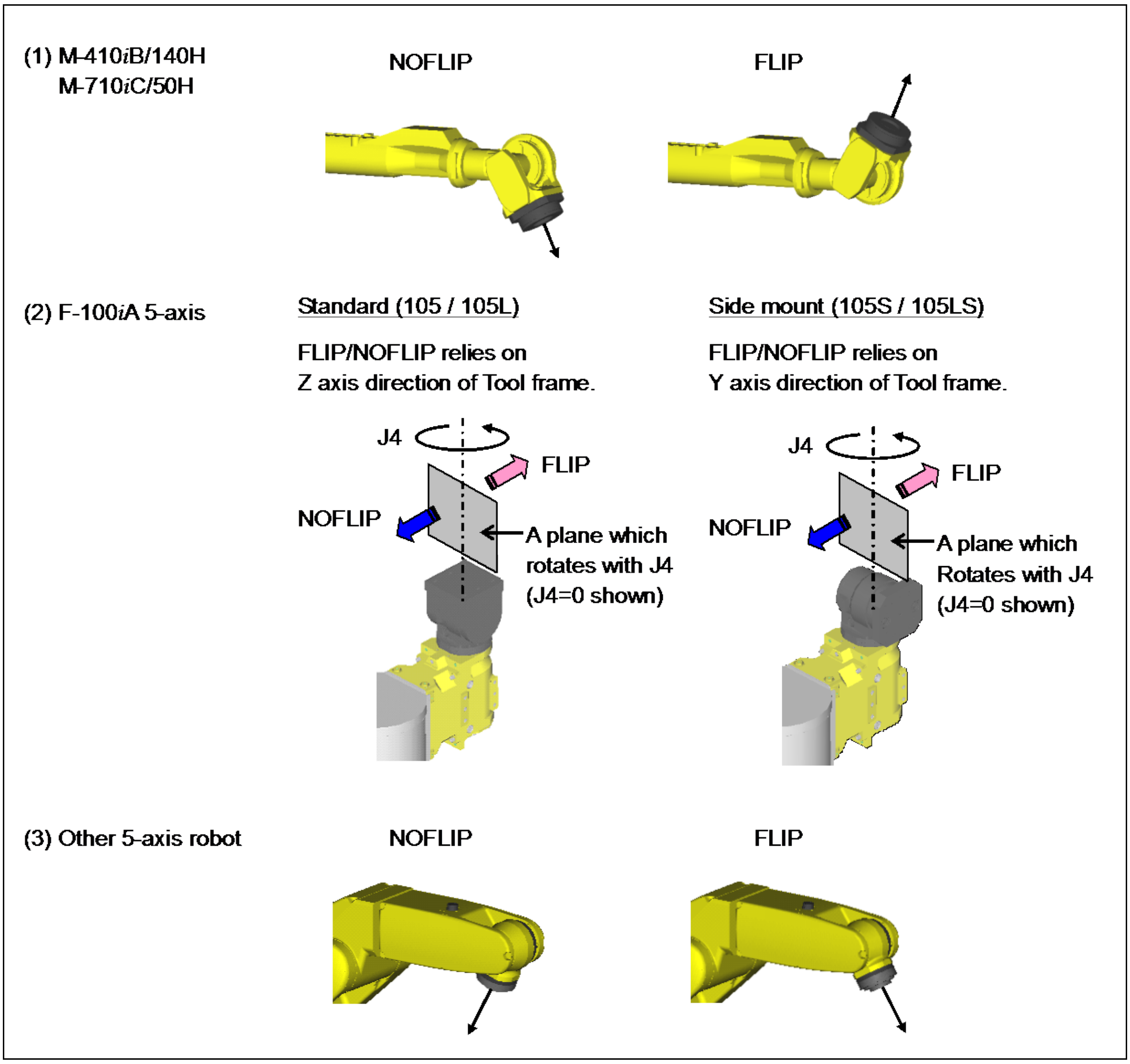


Fig. 4.3.1(f) Definition of configuration for 5-axis robot

3 Group (3) 5-axis robot switches the attitude control method according to the following condition;
(A) The flange surface faces in the vertical direction.
(B) Other than condition (A).
For this reason, 5-axis robot cannot perform a Cartesian motion (Linear motion, Circular motion and Circle Arc motion) if the condition changes from (A) to (B) or from (B) to (A) during the motion. If you teach and start such a motion, an alarm “MOTN-063 Position config change” or “MOTN-310 Pos. Cfg. change2” occurs.

When the servo power turns off due to an emergency stop input etc., sometimes robot axis slip slightly and the condition changes from (A) to (B). Then, you may see the alarm described above. In this case, resume the program as follows;

Step1: Abort the program.
Step2: Jog the robot to recover condition (A).
Step3: Resume the program.

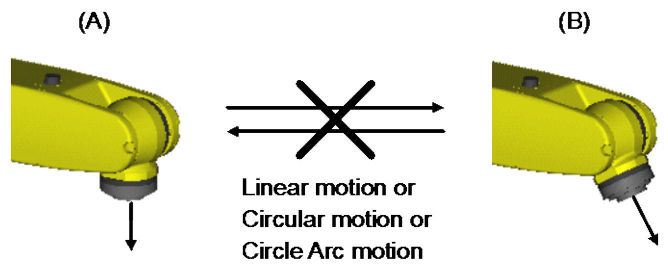


Fig. 4.3.1(g) Prohibition of Cartesian motion switched the attitude control method

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.

UF,	UT,	(X, Y, Z,	W, P, R),	Configuraton
User coordinate system number	Tool coordinate system number	Position	Attitude	Configuration

Fig. 4.3.2 (a) Position data (Cartesian coordinates)

Position and attitude

- The position (x, y, z) represents the three-dimensional position of the tool center point (origin of the tool coordinate system) in the Cartesian coordinate system.
- The attitude (w, p, r) represents angular displacements about the X-axis, Y-axis, and Z-axis in the Cartesian coordinate system.

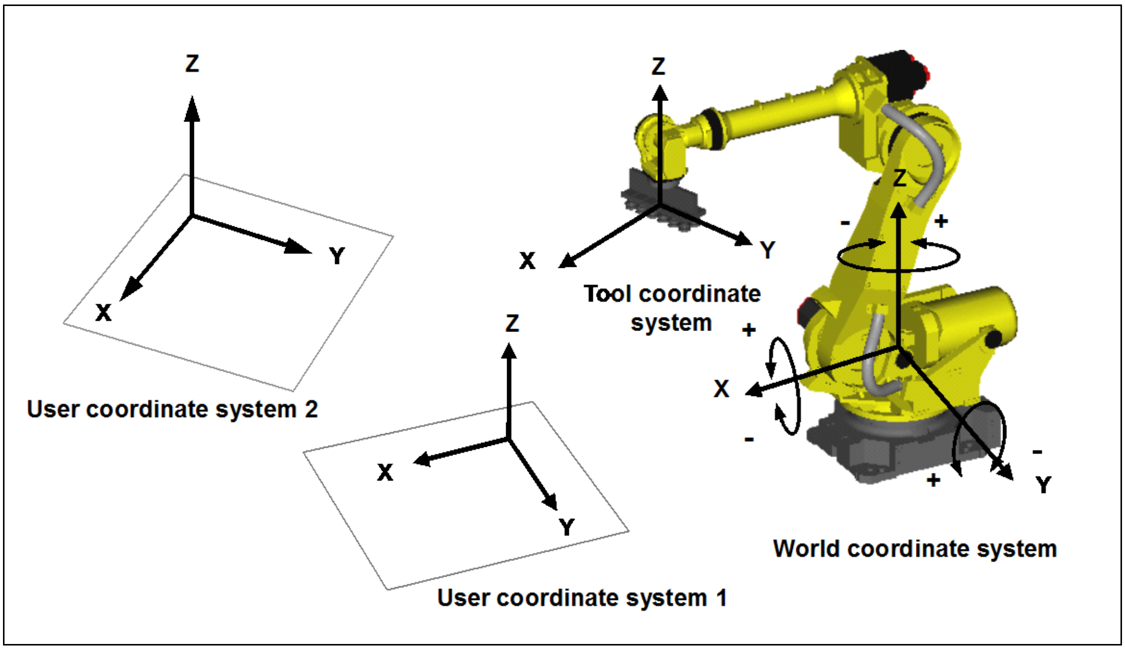


Fig. 4.3.2 (b) World coordinate system/tool coordinate system

Configuration

A configuration represents the attitude of the robot. Several configurations are available which meet the condition of Cartesian coordinates (x, y, z, w, p, r). The Turn Number and Joint Placement of each axis must be specified.

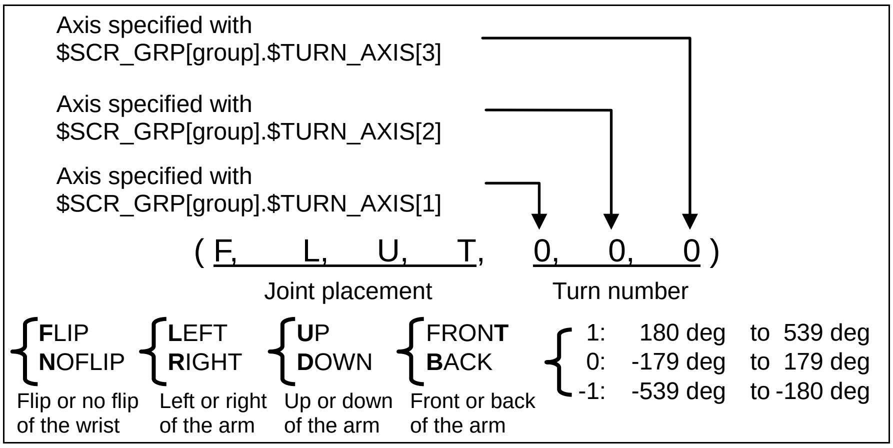


Fig. 4.3.2 (c) Configuration

- 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 point, or to be taking a peculiar attitude. At the singular point, since the configuration can not be decided to one by the specified Cartesian coordinate values, the robot can not move.

- An operation that ends at a singular point cannot be programmed. (In some cases, the most feasible configuration can be selected.) To specify such an operation, define the axial coordinate values.
- During linear or circular motion or circle arc motion, the tool cannot pass through a singular point (the joint placement cannot be changed). In this case, execute a joint motion. To pass through a singular point on the wrist axis, a wrist joint motion (Wjnt) can also be executed.

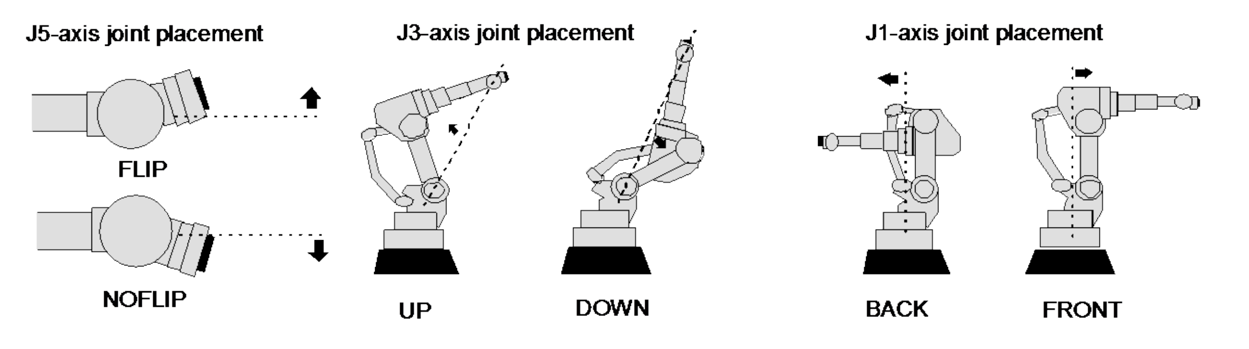


Fig. 4.3.2 (d) Joint placement

Turn number

Turn number represents the number of revolutions of the wrist axis (J4, J5, J6). Each axis returns to the original position after one revolution. So, specify how many turns have been made. Turn number is 0 when each axis is at an attitude of 0.

The turn numbers for up to three axes can be displayed. The axis number to correspond to each field is specified with system variable $SCR_GRP[i].$TURN_AXIS[j] (where i is a group number), as follows:

Left field : Axis number specified with $SCR_GRP[i].$TURN_AXIS[1]
Middle field : Axis number specified with $SCR_GRP[i].$TURN_AXIS[2]
Right field : Axis number specified with $SCR_GRP[i].$TURN_AXIS[3]

When programmed linear motion or circular motion or circle arc 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 for the tool coordinate system, and a number from 1 to 9 for the user coordinate system) specified in the position data does not match the coordinate system number currently selected, the program is not executed for safety, and an alarm is issued.
A coordinate system number is written into position data in position teaching.
To change a coordinate system number after it has been written, use the tool replacement/coordinate replacement shift function.

Tool coordinate system number (UT)

The tool coordinate system number specifies the coordinate system number of a mechanical interface coordinate system or tool coordinate system. Thus, the coordinate system of the tool is determined.

- 0 : The mechanical interface coordinate system is used.
- 1 to 10 : The tool coordinate system of a specified tool coordinate system number is used.
- F : The coordinate system of the tool coordinate system number currently selected is used.

User coordinate system number (UF)

The user coordinate system number specifies the coordinate system number of a world coordinate system or user coordinate system. Thus, the coordinate system of work space is determined.

- 0 : The world coordinate system is used.
- 1 to 9 : The user coordinate system of a specified user coordinate system number is used.
- F : The coordinate system of the user coordinate system number currently selected is used.

Detail position data

To display the detail position data, position the cursor to the position number, then press the F5, "POSITION" key.

```
SAMPLE1
      P[1] UF:0  UT:1              CONF:NUT 000
       X   1500.374   mm  W     40.000   deg
       Y   -242.992   mm  P     10.000   deg
       Z    956.895   mm  R     20.000   deg
      Position Detail
```

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.

(J1, J2, J3, J4, J5, J6, E1, E2, E3)

Main axis — J1, J2, J3
Wrist axis — J4, J5, J6
Additional axis — E1, E2, E3

Fig. 4.3.2 (e) Position data (joint coordinates)

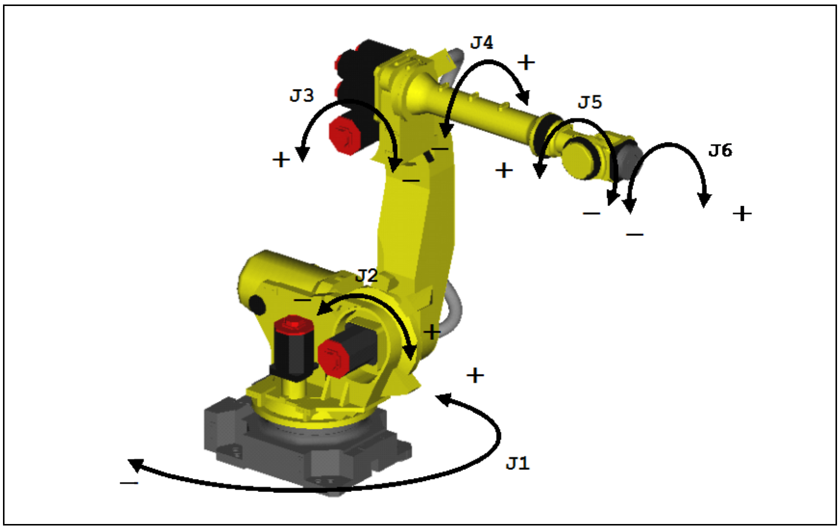


Fig. 4.3.2 (f) Joint coordinate system

Detail position data

Detailed position data is displayed when F5, "POSITION" is pressed. You can chose between Cartesian coordinates and axial coordinates by pressing F5, [REPRE].

```
SAMPLE1
      P[1] UF:0  UT:1
      J1         0.125 deg  J4 -95.000 deg
      J2        23.590 deg  J5   0.789 deg
      J3        30.300 deg  J6-120.005 deg
     Position Detail
```

Position variable and position register

In a motion instruction, position data is represented by a position variable (P[i]) or position register (PR[i]). Usually, a position variable is used.

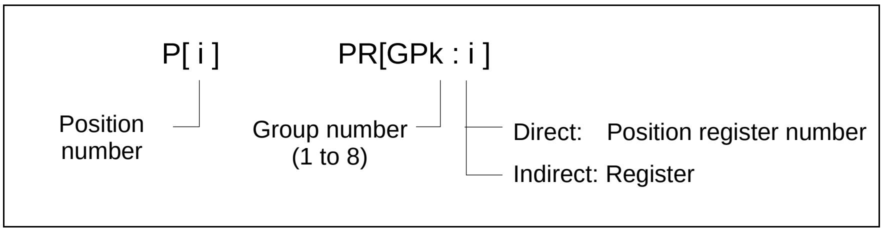


Fig. 4.3.2 (g) Position variable and position register

Example

```
1: J  P[12] 30% FINE
2: L  PR[1] 300mm/s CNT50
3: L  PR[R[3]] 300mm/s CNT50
```

- Position variable

The position variable is the variable usually used to hold position data. In motion instruction teaching, position data is automatically saved.

When Cartesian coordinates are taught, the following Cartesian coordinate system and coordinate system number are used:

- Coordinate system of the tool coordinate system number currently selected (UT = 1 to 10)
- Coordinate system of the user coordinate system number currently selected (UF = 0 to 9)

In playback, the following Cartesian coordinate system and coordinate system number are used:

- Coordinate system with the specified tool coordinate system number (UT = 1 to 10)
- Coordinate system with the specified user coordinate system number (UF = 0 to 9)

- Position register

The position register functions as a general-purpose register for holding position data. (For position teaching using a position register, see Section 7.4.)

When Cartesian coordinates are taught, the following Cartesian coordinate system and coordinate system number are used:

- Coordinate system of the tool coordinate system number currently selected (UT = F)
- Coordinate system of the user coordinate system number currently selected (UF = F)

In playback, the following Cartesian coordinate system and coordinate system number are used:

- Coordinate system of the tool coordinate system number currently selected (UT = F)
- Coordinate system of the user coordinate system number currently selected (UF = F)

- 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/sec CNT50
```

4.3.3 Feed Rate

The feed rate specifies the speed at which the robot moves. During program execution, the feed rate is controlled by feed rate overriding. A feed rate override value of 1% to 100% can be used.
The unit used to specify a feed rate depends on the motion format taught with a motion instruction.

> **NOTE**
> The programmed traveling speed cannot exceed the allowable range of the robot. If a speed exceeding the range is programmed, a warning alarm would be issued.

J P[1] 50% FINE

When the motion type is joint, a feed rate is specified as the following:

- A percentage from 1% to 100% of the maximum feed rate 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 or circle arc motion, specify a feed rate as follows:

- When the unit is mm/sec, specify a feed rate from 1 to 2000 mm/sec.
- When the unit is cm/min, specify a feed rate from 1 to 12000 cm/min.
- When the unit is inch/min, specify a feed rate from 0.1 to 4724.4 inch/min.
- When the unit is sec, specify the value from 0.1 to 3200 sec as the time took for motion.
- When the unit is msec, specify the value from 1 to 32000 msec 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 3200 sec as the time took for motion.
- When the unit is msec, specify the value from 1 to 32000 msec as the time took for motion.

Specifying the feed rate with a register

The feed rate can be specified with a register. This allows the user to specify the feed rate for an operation instruction after calculating the feed rate using a register. The feed rate can also be specified externally, using group input (GI) or data transfer, for example.

> ⚠ **CAUTION**
> This function allows the user to change the feed rate 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 feed rate is specified with a register

- Joint J P[1] R[i]% FINE
- Linear L P[1] R[i]mm/sec FINE

- Circular C P[1]
  P[2] R[i]mm/sec FINE
- Circle Arc A P[1] R[i]mm/sec FINE
- Pallet operation instruction
  J PAL_1[A_1] R[i]% FINE
  J PAL_1[BTM] R[i]% FINE
  J PAL_1[R_1] R[i]% FINE

NOTE
The pallet operation instruction is a software option of palletizing.

- Operation group instruction
  Asynchronous operation group
  GP1 JP[1] R[i]% FINE
  GP2 JP[1] R[i]% FINE

NOTE
The operation group instruction is a software option of multimotion.

The feed rate 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 feed rate for an operation addition instruction is not supported.
In program editing, a range check is not performed on the feed rate (register value).
The feed rate (register value) is not automatically converted when the feed rate unit is changed.
If the feed rate 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 feed rate (integer/real), an alarm is generated during execution.

Unit	Allowable range	
%	1 to 100	Integer
sec	0.1 to 3200.0	Real/effective up to the first decimal place.
msec	1 to 32000	Integer
mm/sec	1 to 2000	Integer
cm/min	1 to 12000	Integer
inch/min	0.1 to 4724.2	Real/effective up to the first decimal place.
deg/sec	1 to 272	Integer

The allowable range (maximum value) differs depending on the robot type.
Read-ahead can be enabled.
If the feed rate 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 feed rate 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 feed rate is enabled, it is necessary to take appropriate measures such as interlocking or not changing the value of the register used for the feed rate 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. Two positioning path modes are available:

- FINE positioning path
- CNT positioning path

FINE positioning path J P[i] 50% FINE

When the FINE positioning path is specified, the robot stops at a target point before moving to the next target point.

CNT positioning path J P[i] 50% CNT50

When the CNT positioning path is specified, the robot approaches a target point but does not stop at the point and moves to the next point.

How closely the robot must approach a target point can be defined by specifying a value from 0 to 100. Register is also used to set the value for indirect expression. The range of register index is from 1 to 255.

When 0 is specified, the robot moves the nearest path to the destination position but moves to the next target point without stopping at the target point. When 100 is specified, the robot moves along the farthest path to the target point because the robot does not decelerate near the target point and it starts to move to the next target point soon.

NOTE

1 When an instruction such as a wait instruction is taught, the robot stops at the target point to execute that instruction.

2 Several short-distance, high-speed motions that are performed continuously with CNT specified may be decelerated, even if the specified CNT value is 100.

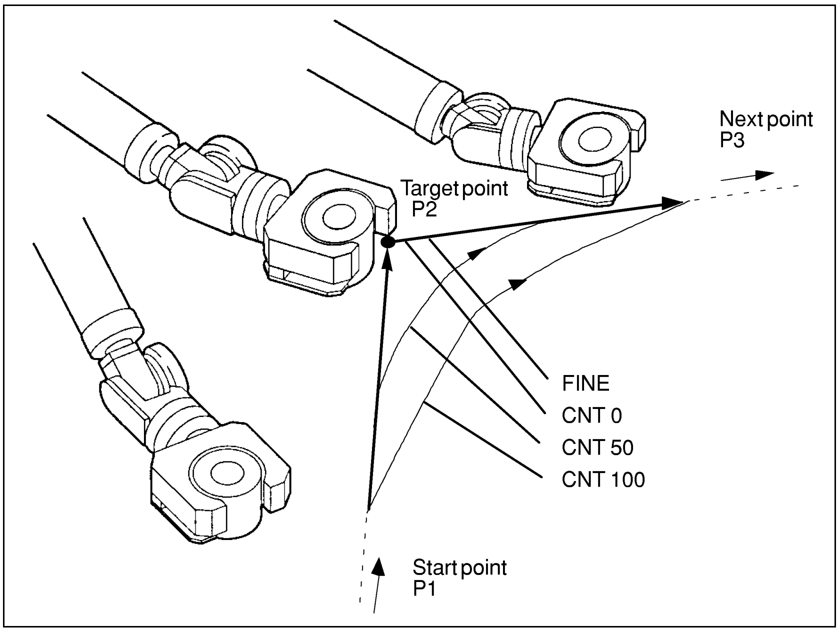


Fig. 4.3.4 Robot motion path using continuous termination type

4.3.5 Additional Motion Instructions

An additional motion instruction causes the robot to perform a particular job. The following additional motion instructions are available:

- Wrist joint motion instruction (Wjnt)
- Acceleration override instruction (ACC)
- Skip instruction (Skip,LBL[i])
- High-speed skip instruction (Skip,LBL[i],PR[j])
- 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 9.5, TIME BEFORE FUNCTION)
- Break instruction (BREAK)
- Orient Base instruction (ORNT_BASE)

NOTE
Only one additional motion instruction can be taught to the motion instruction for the passing point of the circular motion. If you want to teach two or more additional motion instruction to the passing point of the circular motion, teach multiple linear motion instructions to approximate the circular motion.

NOTE
About the additional motion instruction which can be taught to the Circle Arc motion instruction, refer to the Subsection “9.18.14 Available Motion options”.

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.

```
Motion Modify   1
1 No option
2 Wrist Joint
3 ACC
4 Skip,LBL[]
5 BREAK
6 Offset/Frames
7 Offset,PR[ ]
8 --next page—
```

```
Motion Modify   2
1 Incremental
2 ORNT_BASE LDR
3 ORNT_BASE UF
4 ORNT_BASE
5 Tool_Offset
6 Tool_offset,PR[
7 Independent EV
8 --next page--
```

```
Motion Modify   3
1 Simultaneous EV
2 TIME BEFORE
3 Skip,LBL,PR
4 TIME AFTER
5 DISTANCE BEFORE
6 PTH

8 --next page--
```

Wrist joint motion instruction L P[i] 50mm/sec FINE Wjnt

```
Motion Modify   1
1 No option
2 Wrist Joint
3 ACC
4 Skip,LBL[]
5 BREAK
6 Offset/Frames
7 Offset,PR[ ]
8 --next page--
```

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 or circle arc 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
1 No option
2 Wrist Joint
3 ACC
4 Skip,LBL[]
5 BREAK
6 Offset/Frames
7 Offset,PR[ ]
8 --next page--
```

This instruction specifies the percentage of the acceleration/deceleration rate during motion. Basically, this function is for making motion slower. When the acceleration override is reduced, acceleration time will be long (Acceleration and deceleration are done slowly). To perform a potentially dangerous operation such as hot water scooping, use a value less than 100%. When acceleration override is raised, acceleration time will be short (Acceleration and decelerate are done quickly).
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 100%. Register is also used to set the acceleration override for indirect expression.

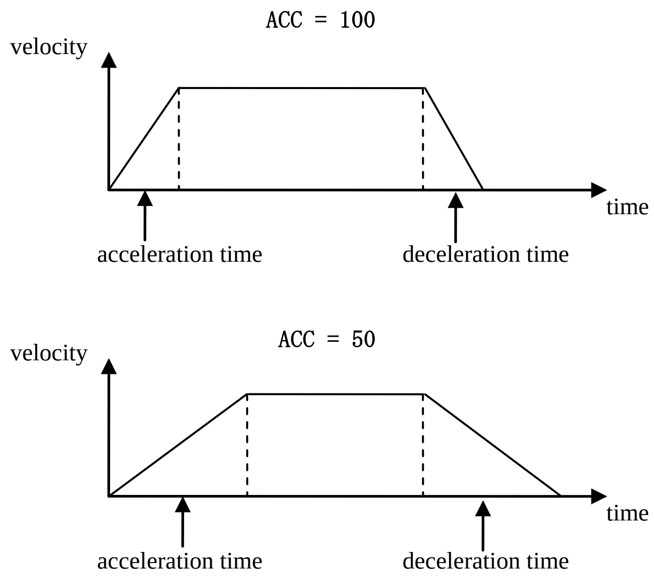


Fig. 4.3.5 (a) Acceleration override

> ⚠ **CAUTION**
> Over 100% acceleration may cause awkward motion or vibration. Since large current instantaneously flows to the primary power source, the input voltage may drop depending on equipment power capacity, and this may cause a servo alarm such as power alarm, position error excess, and servo amplifier low voltage. If such an alarm occurs, either reduce the acceleration/deceleration override value or delete the accelerate/deceleration override instruction.
> And, over 100% acceleration increases the load to the robot arm. It may cause the failure of the robot mechanical parts, premature breakdown of reducer and life shortage of reducer.

Skip instruction

- **SKIP CONDITION [I/O] = [value]**
- **J P[1] 50% FINE Skip,LBL[3]**

```
Motion Modify   1
1 No option
2 Wrist Joint
3 ACC
4 Skip,LBL[]
5 BREAK
6 Offset/Frames
7 Offset,PR[ ]
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 while the robot is moving to a target point, the robot cancels the motion and program execution proceeds to the program statement on the next line. If the skip condition is not satisfied, program execution skips (jumps) to the line of the branch destination label after completion of the robot motion.
The skip condition instruction specifies, in advance, a skip condition (condition for executing a skip instruction) to be used with it. Before a skip instruction can be executed, a skip condition instruction must be executed. A skip condition once specified is valid until the execution of the program is completed, or the

next skip condition instruction is executed. (For the branch instructions, see Section 4.7. For the skip condition instruction, see Section 4.9.)

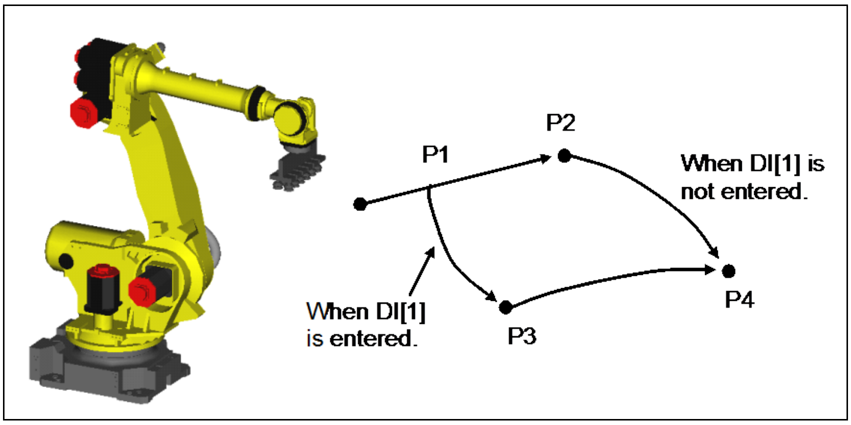


Fig. 4.3.5 (b) Skip instruction

Example

```
1: SKIP CONDITION DI[1] = ON
2: J  P[1] 100% FINE
3: L  P[2] 1000mm/sec FINE Skip, LBL[1]
4: J  P[3] 50% FINE
5: LBL[1]
6: J  P[4] 50% FINE
```

High-speed skip instruction

- **SKIP CONDITION [I/O] = [value]**
- **L P[1] 100mm/sec FINE**
  Skip, LBL[3], PR[5] = LPOS

```
Motion Modify   1
1 No option
2 Wrist Joint
3 ACC
4 Skip,LBL[]
5 BREAK
6 Offset/Frames
7 Offset,PR[ ]
8 --next page--
```

```
Motion Modify   2
1 Incremental
2 Tool_Offset
3
Tool_offset,PR[]
4 Independent EV
5 Simultaneous EV
6 TIME BEFORE
7 Skip,LBL,PR
8 --next page--
```

A high-speed 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 high-speed skip instruction) to be used with it. Before a high-speed 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.)

- Difference between High-speed skip instruction and Skip instruction

(1) When a high-speed skip instruction is used, the position of the robot when the skip conditions are met is stored in programmed position registers.

(2) At the moment when the skip conditions are met, the robot decelerates and stops when an ordinary skip instruction is used, on the other hand, the robot stops quickly by developing the maximum torque of the motor when a high-speed skip instruction is used. After the robot stops, the program statement on the next line is executed whichever instruction is used.

- Use method

Program teaching

a) Teaching skip conditions
The skip conditions for the high-speed skip instruction are taught in the same way as the ordinary skip instruction.

b) Teaching a high-speed skip instruction (an additional operation instruction)
Select the high-speed skip instruction “Skip, LBL, PR” from the additional operation instruction menu.

c) Specify the label, position register, and position storage format.

```
Skip, LBL[10], PR[5]=LPOS
                    or JPOS
```

[Sample program]

```
              :
 8: SKIP CONDITION DI[3]=ON
              :
10: L P[2] 500mm/sec FINE
11: L P[3] 100mm/sec
  :  SKIP, LBL[10], PR[5]=LPOS
              :
              :
30: LBL[10]
```

Explanation of the execution example

When DI[3] is turned on during execution of the 11th line, the current position is stored in a form of Cartesian coordinates.
When DI[3] is not turned on during execution of the 11th line, a branch to LBL[10] is made after the execution of the 11th line ends. In this case, no position data is stored in PR[5].

- Limitations and notes

(1) Position read error
As the programmed operation speed is slower, the position read accuracy under skip conditions becomes higher. (As a guideline, an error of about 1.5 mm is generated for 100 mm/sec. The error is proportional to the speed.)

(2) Motion speed limit
To decrease impact of stopping when skip condition is satisfied, speed limit is set for the motion with high-speed skip instruction. The speed limit is adjusted for each robot models. When taught speed exceeds the limit, speed limit works automatically. When speed limit works, the following warning is displayed.
MOTN-560 Hspd.skip speed limit (G:group number)

OFFSET instruction

Offset,PR[2] (UFRAME [1])
J P[1] 50% FINE Offset

```
Motion Modify    1
1 No option
2 Wrist Joint
3 ACC
4 Skip,LBL[]
5 BREAK
6 Offset/Frames
7 Offset,PR[ ]
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 4.13, ”FRAME INSTRUCTIONS.) When it is not specified, the user frame (UF) being selected now is used. (See Section 4.11, ”OFFSET CONDITION INSTRUCTION”.)

The setting values of the tool frame number (UT) and the configuration (CONF) are ignored.

> ⚠ **CAUTION**
> If teaching is made by joint coordinates, changing the user coordinate system does not affect the position variables and position registers. However, note that both position variables and registers are affected by the user coordinate systems when the robot is taught in the Cartesian format.

When you 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 3.16 SETTING THE GENERAL ITEMS.)

When Ignore Offset command is set to ENABLED, the current position is directly taught as the positional information (The prompt message is not displayed) and the robot stops at the teaching position even if the OFFSET instruction is executed.
When the offset amount is changed after the program is paused while the OFFSET instruction is in progress, this change is reflected to the motion after the program is resumed. But, when you change the number of a position register in the OFFSET CONDITION instruction, this change is not reflected in the motion.
The robot moves to the offset position at the backward execution. (See Subsection 6.3.2,Step Test.)
This is the same as the following explanation for the direct offset condition instruction.

Direct offset condition instruction

J P[1] 50% FINE Offset,PR[2]

```
Motion Modify   1
1 No option
2 Wrist Joint
3 ACC
4 Skip,LBL[]
5 BREAK
6 Offset/Frames
7 Offset,PR[ ]
8 --next page--
```

The direct offset condition instruction alters positional information by the offset amount directly specified in the position register without using the offset condition specified in the OFFSET CONDITION instruction. The reference frame is specified by the number of the user frame currently selected.

> ⚠ **CAUTION**
> If teaching is made by joint coordinates, changing the user coordinate system does not affect the position variables and position registers. However, note that both position variables and registers are affected by the user coordinate systems when the robot is taught in the Cartesian format.

When you change or edit the motion instruction with the direct offset condition option, you can teach the positional information minus the offset amount.
When you teach or edit the motion instruction with the direct offset condition option, the following prompt message is displayed:

- Subtract offset data from current pos?
  - "Yes" Subtract the offset data from the taught position.
  - "No" The positional information from the directly taught position.

When the positional information is manually edited with the numerical keys, you can not teach the positional information minus the offset amount.
Moreover, even if the position teaching by which the subtracted offset amount is effective, the current position will be taught as it is in the following cases:

- The specified position register is non-initialized.
- The position register number used by direct offset condition instruction is non-initialized.
- Ignore Offset command is set to ENABLED. (See Section 3.16 SETTING THE GENERAL ITEMS.)

When Ignore Offset command is set to ENABLED, the current position is directly taught as the positional information (The prompt message is not displayed) and the robot stops at the teaching point even if the offset instruction is executed.

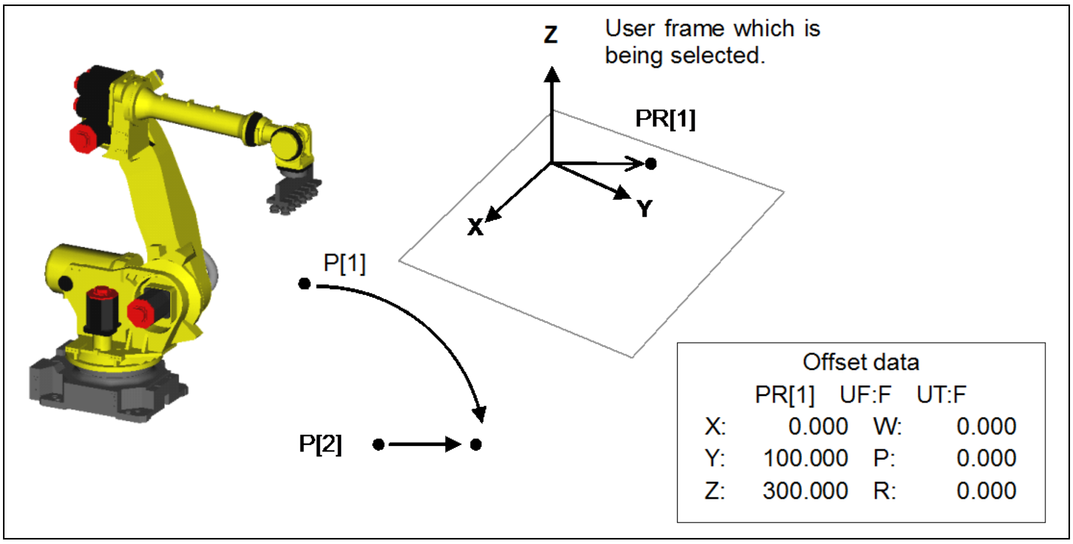


Fig. 4.3.5 (c) Offset instruction

Example 1
```
1:    OFFSET CONDITION PR[1]
2: J  P[1] 100% FINE
3: L  P[2] 500mm/sec FINE Offset
```

Example 2
```
1: J  P[1] 100% FINE
2: L  P[2] 500mm/sec FINE Offset, PR[1]
```

Tool offset instruction

TOOL_OFFSET CONDITION PR[2] (UTOOL[1])
J P[1] 50% FINE Tool_offset

```
Motion Modify   2
1 Incremental
2 Tool_Offset
3
Tool_Offset,PR[]
4 Independent EV
5 Simultaneous EV
6 TIME BEFORE
7 Skip,LBL,PR
8 --next page—
```

A tool offset instruction moves the robot to the position shifted from the target position, recorded in the position data, by the offset specified in the tool offset conditions. The condition when the offset is applied is specified by a tool offset condition instruction.

A tool offset condition instruction specifies the offset condition used in a tool offset instruction. Execute a tool offset condition instruction before executing the corresponding tool offset instruction. Once the tool offset condition has been specified, it remains effective until the program terminates or the next tool offset condition instruction is executed.

Note the following when specifying tool offset conditions.

- The position register specifies the direction in which the target position shifts, as well as the amount of the shift.
- The tool coordinate system is used for specifying offset conditions.
- When the number of a tool coordinate system is omitted, the currently selected tool coordinate system is used.

When a motion statement which includes a tool offset instruction is taught or a certain position is modified, the position to which the offset is not to be applied can be taught.

When a motion statement which includes a tool offset instruction is taught or a certain position is modified, the system prompts the operator to respond to enter data in response to the following messages.

- Subtract tool offset data?
  - Pressing the "YES" soft key subtracts the tool offset from the position data and the robot is taught the new position.
  - Pressing the "NO" soft key stores the current position as the position data.
- Enter PR index of tool offset data?
  - Specify the position-register number specified by the tool offset condition instruction.
- Enter tool no. of tool offset data?
  - Specify the number of the tool coordinate system in which the offset is to be specified.

When the position data is manually modified with the numeric keys, the position is taught without subtracting the offset.

Even when teaching of the position from which the offset is subtracted is enabled, the current position is stored in the following cases.

- When the specified position register has not yet been initialized
- When Ignore Tool_offset is set to ENABLED. (See Section 3.16 SETTING THE GENERAL ITEMS.)

When Ignore Tool_offset is set to ENABLED, the current position is taught as position data (no prompt messages are output) and the robot is moved to the taught position, even if a tool offset instruction is executed.

When the robot is temporarily stopped during the execution of a tool offset instruction and the shift distance is modified, the modified distance is used in the resumed movement. When a position register number specified by a tool offset condition instruction is modified, the modified number is not used.
In backward execution (See Subsection 6.3.2, Step Test), the robot is moved to the position to which the offset has been applied. This also applies to the direct tool offset instruction, described next.

Direct tool offset instruction

J P[1] 50% FINE Tool_Offset, PR[2]

```
Motion Modify   2
1 Incremental
2 Tool_Offset
3
Tool_Offset,PR[]
4 Independent EV
5 Simultaneous EV
6 TIME BEFORE
7 Skip,LBL,PR
8 --next page--
```

The robot moves according to the offset stored in the specified position register, ignoring the tool offset conditions specified by the tool offset condition instruction. The currently selected tool coordinate system is used.
When a motion statement which includes a direct tool offset instruction is taught or a certain position is modified, the position to which the offset is not to be applied can be taught.
When a motion statement which includes a direct tool offset instruction is taught or a certain position is modified, the system prompts the operator to enter data in response to the following messages.

- Subtract tool offset data?
  - Pressing the "YES" soft key subtracts the tool offset from the position data and the robot is taught the new position.
  - Pressing the "NO" soft key stores the current position as position data.

When the position data is manually modified with the numeric keys, the position is taught without subtracting the offset.
When teaching of the position from which the offset is subtracted is enabled, the current position is stored in the following cases.

- When the specified position register has not yet been initialized
- When the direct tool offset instruction has not specified the number of a position register
- When Ignore Tool_offset is set to ENABLED. (See Section 3.16 "SETTING THE GENERAL ITEMS".)

When Ignore Tool_offset is set to ENABLED, the current position is taught as position data (no prompt messages are output) and the robot is moved to the taught position even if a tool offset instruction is executed.

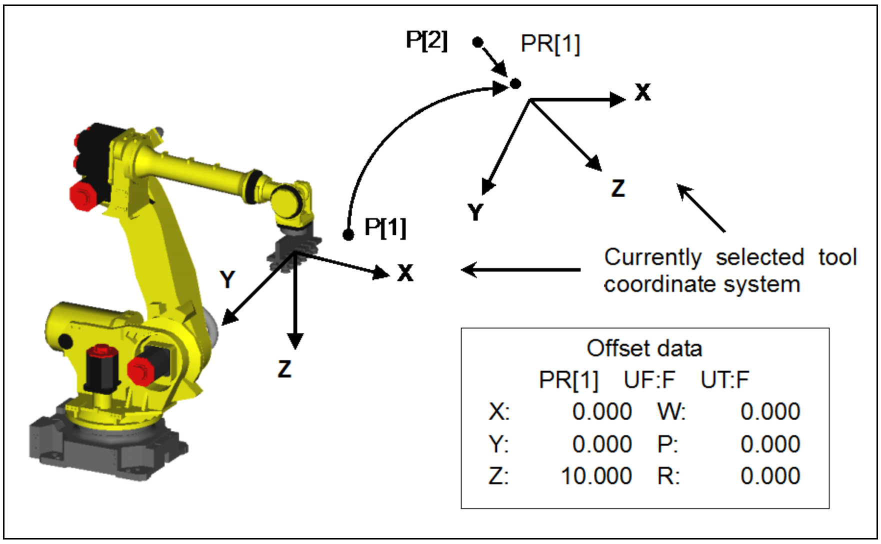


Fig. 4.3.5 (d) Tool offset instruction

Example 1
```
1:   TOOL_OFFSET CONDITION PR[1]
2: J P[1] 100% FINE
3: L P[2] 500mm/sec FINE Tool_Offset
```

Example 2
```
1: J P[1] 100% FINE
2: L P[2] 500mm/sec FINE Tool_Offset, PR[1]
```

Incremental instruction

J P[1] 50% FINE INC

```
Motion Modify    2
1 Incremental
2 Tool_Offset
3
Tool_Offset,PR[]
4 Independent EV
5 Simultaneous EV
6 TIME BEFORE
7 Skip,LBL,PR
8 --next page--
```

The incremental instruction uses the positional data in the motion instruction as the incremental amount from the current position, and causes the robot to move to the destination position that the incremental amount is added to the current position. This means that the incremental motion amount from the current position is recorded in the positional data in the motion instruction.
The incremental condition is specified by the following elements:

- When the positional data is joint frame value, the incremental amount of each axis is applied.
- When the positional variable (P[]) is used as the positional data, the reference user frame is specified by the number of the user frame which is specified in the positional data. However, the frame is verified. (Cartesian coordinate system reference)
- When the position register is used as the position data, the reference frame is the current user frame.
- When the INC instruction is used with the OFFSET instruction, the type of the positional data in the motion instruction should be corresponding to the type of the positional register for the offset. In this case, the offset amount is used as the offset amount of the specified incremental amount.

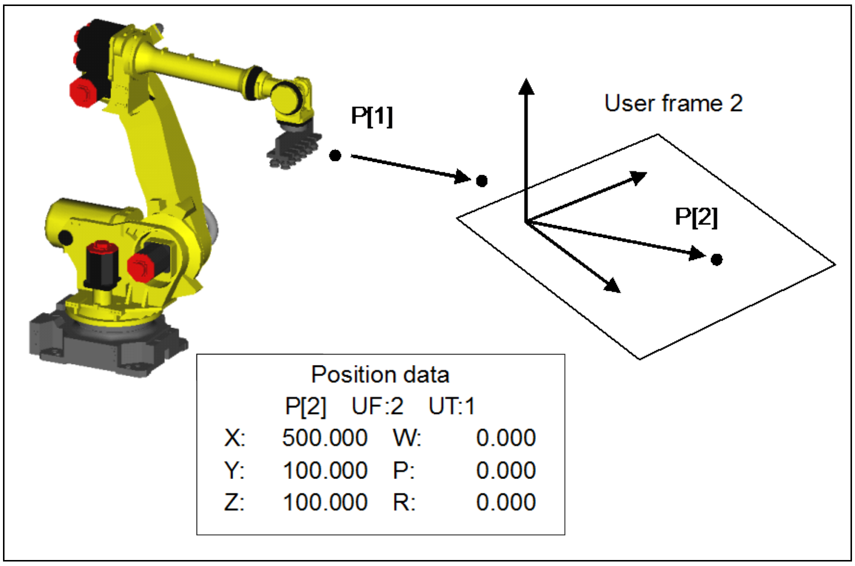


Fig. 4.3.5 (e) Incremental instruction

Example
```
1: J  P[1] 100% FINE
2: L  P[2] 500mm/sec FINE INC
```

Note the following when teaching the incremental instruction (See Subsection 5.3.4, Teaching an Additional 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 change 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   2
1 Incremental
2 Tool_Offset
3
Tool_Offset,PR[]
4 Independent EV
5 Simultaneous EV
6 TIME BEFORE
7 Skip,LBL,PR
8 --next page–
```

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   2
1 Incremental
2 Tool_Offset
3 Tool_Offset,PR[
4 Independent EV
5 Simultaneous EV
6 TIME BEFORE
7 Skip,LBL,PR
8 --next page—
```

The additional axis speed instruction (asynchronous) moves the robot asynchronously with the additional axis.
When this instruction is used, the robot and the additional axis start moving at the same time, but stop at different times because they are not synchronized.
The extended axis speed is specified as a ratio (1% to 100%) to the maximum travel speed of the extended axis.
If a motion statement is not accompanied with either extended axis speed instruction, the extended axis moves in synchronization with the speed of the robot.

Path instruction

J P[1] 50% Cnt10 PTH

```
Motion Modify   3
1 TIME AFTER
2 DISTANCE BEFORE
3 PTH
4
5
6
7
8 --next page--
```

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 Cnt is smaller).
When using this function, note the following:
In the following cases, use of the PTH instruction may actually incur a longer cycle time:
Before using this function, therefore, confirm its effect.

- A large Cnt value is specified in a motion statement.
- A motion statement causes the robot to move through a long distance.
- Successive Cnt motion statements appear.

> ⚠ **CAUTION**
> Some motion instructions that use the PTH switch might cause jerky motion or vibration. If the motion is attached to PTH has a vibration, delete the PTH motion option.

BREAK instruction

L P[1] 2000mm/sec CNT100 <u>BREAK</u>

By using BREAK instruction, user can have the robot waiting at taught position even if the term type before WAIT instruction is CNT.

When BREAK instruction is added to the motion instruction with CNT before WAIT instruction, the next motion does not start until the condition of WAIT is satisfied. Therefore the robot moves to the taught position until then and the path of the corner is changed depending on WAIT time. If WAIT time is long enough, the robot waits on the taught position until the next motion starts.

Example：With BREAK instruction

```
1: L P[1] 2000mm/sec FINE
2: L P[2] 2000mm/sec CNT100 BREAK
3:    WAIT DI[1]=ON or WAIT x(sec)
4: L P[3] 2000mm/sec FINE
```

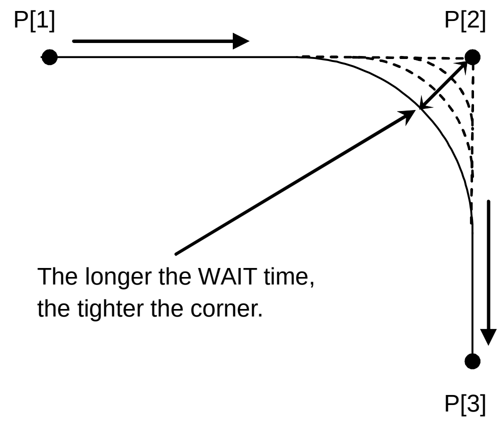


Fig. 4.3.5(f) BREAK instruction

Reference

Normally, when there is WAIT instruction next to the motion instruction with CNT, the robot waits on the corner until the condition of WAIT is satisfied. The path of the corner is constant regardless of WAIT time.

Example：Without BREAK instruction

```
1: L P[1] 2000mm/sec FINE
2: L P[2] 2000mm/sec CNT100
3:     WAIT DI[1]=ON or WAIT x (sec)
4: L P[3] 2000mm/sec FINE
```

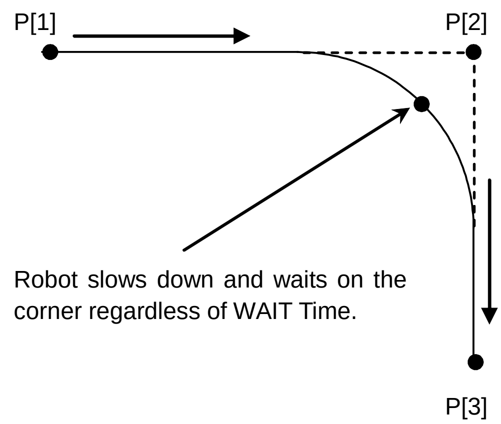


Fig. 4.3.5(g) BREAK instruction (Without BREAK instruction)

ORNT_BASE instruction

C P[1]
P[2] 100cm/min CND100 ORNT_BASE;

```
Motion Modify   2
1  Incremental
2  ORNT BASE LDR
3  ORNT BASE UF
4  ORNT BASE
5  Tool_Offset
6  Tool_offset,PR
7  Independent EV
8 --next page--
```

> **NOTE**
> ORNT_BASE instruction is available in the version 7DF1/15 or later.

ORNT_BASE instruction is designed to improve the performance of circular motion with up-and-down on its path. This instruction is suitable for saddle-shaped welding shown in Fig. 4.3.5(h).

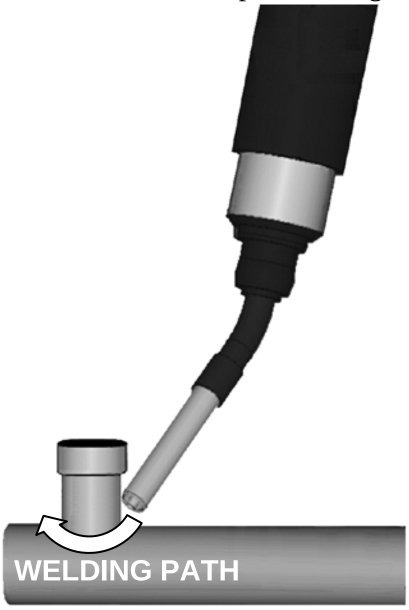


Fig. 4.3.5(h) Saddle-shaped welding, which is suitable for ORNT_BASE.

Without this instruction, tool orientation deviates between taught points as shown in “(A) Without ORNT_BASE instruction” in Fig. 4.3.5(i)

By using this instruction, tool orientation behaves as shown in “(B) With ORNT_BASE instruction” in Fig. 4.3.5(i) and that improves the quality of welding.

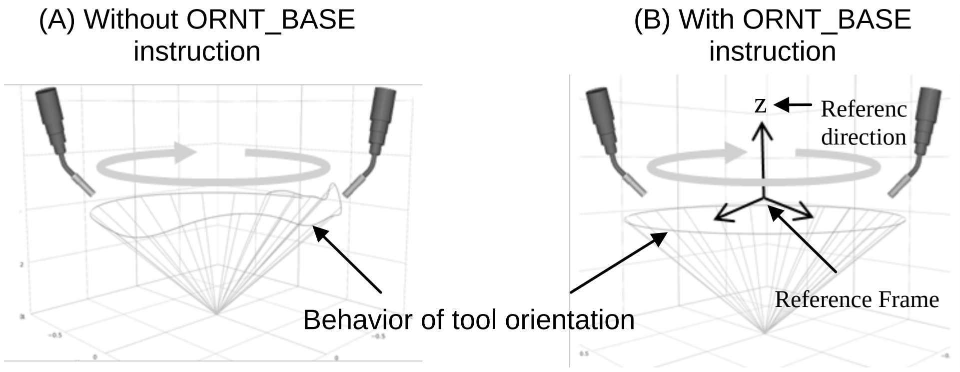


Fig. 4.3.5(i) Behavior of tool orientation, with and without ORNT_BASE instruction.

ORNT_BASE instruction is used to specify the reference direction of orientation for circular-motion. The reference direction is defined by “Reference frame” and “Direction (x, y or z)”. Example of reference direction is shown in “(B) With ORNT_BASE instruction” in Fig. 4.3.5(i).

> ⚠ **CAUTION**
> The orientation of tool might unexpectedly weave when inappropriate reference direction is specified for ORNT_BASE instruction. Please make sure to specify a correct reference direction.

The specifications of ORNT_BASE are written below.

- This instruction is only for Circular and Circular-Arc motion.
- Available for both stationary and non-stationary.
- Available for both coord and non-coord motion.

- Three types of instruction can be specified as below.

ORNT_BASE:

Reference Frame : World Frame
Direction : z axis

ORNT_BASE UF [frm_idx, dir_idx]:

frm_idx: Reference Frame (0 : World Frame, ELSE : User Frame)
dir_idx: Direction (x, y, z)

ORNT_BASE LDR [frm_idx, dir_idx]: (Only available for coord motion)

frm_idx: Leader’s Reference Frame (0 : Leader’s Flange Frame, ELSE : Leader Frame)
dir_idx: Direction (x, y, z)

- Usually, the tool orientation will become stable by using ORNT_BASE(no suffix) instruction. If it doesn't work, use ORNT_BASE UF or ORNT_BASE LDR instruction. ORNT_BASE LDR is only available for coordinated motion, but not always necessary.

- It order to show the ORNT_BASE instruction in Additional Motion Instructions by setting the following system variable.

$CRCFG.$SW_ORNTBASE = 0 : Hide all ORNT_BASE instructions.
$CRCFG.$SW_ORNTBASE = 1 : Show only ORNT_BASE (no suffix) instruction
$CRCFG.$SW_ORNTBASE = 2 : Show all ORNT_BASE instructions.

Default value of this system variable is as below.

Arc Tool	:	$CRCFG.$SW_ORNTBASE = 1
Except Arc Tool	:	$CRCFG.$SW_ORNTBASE = 0

- Positioning Path must be FINE in the following conditions.
  (1) If the reference frame or direction changes at the next motion line.
  (2) At the end of sequence of ORNT_BASE-motion.
  (3) Before the beginning of sequence of ORNT_BASE-motion.

- In a sequence of continuous motion, ORNT_BASE-motion and non-ORNT_BASE-motion cannot be mixed up.

  Use ORNT_BASE as follows;

  [Sample program]

```
1:J P[1] 100% FINE
2:C P[2]
 :  P[3] 20cm/min CNT100 ORNT_BASE
3:C P[4]
 :  P[5] 20cm/min CNT100 ORNT_BASE
4:C P[6]
 :  P[7] 20cm/min FINE ORNT_BASE
5:J P[8] 100% FINE
```

- ORNT_BASE instruction cannot be used with following functions.
  - ✓ Line Tracking
  - ✓ MP/RPM
  - ✓ Universal Sensor Interface
  - ✓ TAST
  - ✓ AVC
  - ✓ Auto Error Recovery
  - ✓ Remote TCP

- Dynamic User Frame cannot be specified as Reference Frame.

- The following alarm messages can be displayed related to ORNT_BASE instruction.

- INTP-204 Invalid value for index	: Non-existent frame number is used.
- CD-008 No leader	: ORNT_BASE LDR is taught without COORD
- MOTN-591 Prev. Term-type not FINE(ORNT)	: CND-motion is used before the beginning of ORNT_BASE-motion.
- MOTN-592 ORNT_BASE must be with C/A	: ORNT_BASE is used with Joint or Linear motion.
- MOTN-593 ORNT_BASE not supported	: Unsupported option is used with ORNT_BASE.

4.4 PALLETIZING INSTRUCTIONS

Palletizing is a function for orderly stacking of workpieces by only teaching several representative points. For the detail of palletizing function, refer to chapter “10. PALLETIZING FUNCTION”.

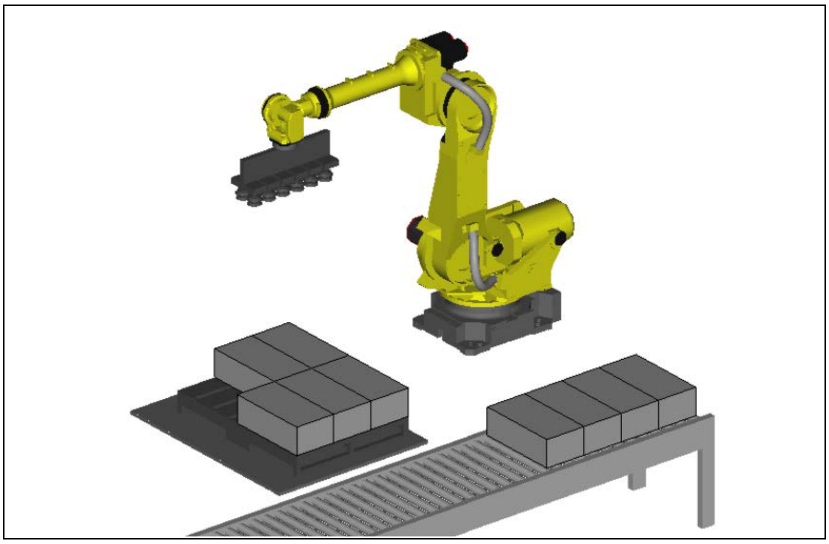


Fig. 4.4 Palletizing

The following palletizing instructions are available:

- Palletizing instruction PALLETIZING-B_i, BX_i, E_i, EX_i
- Palletizing motion instruction J PAL_i [BTM] 100 % FINE
- Palletizing end instruction PALLETIZING-END_i

4.4.1 Palletizing Instruction

Based on the value held in the palletizing register, the palletizing instruction calculates the position of the current stack point from a stacking pattern and the position of the current path from a path pattern. It then writes the found values into the position data of a palletizing motion instruction.

```
PALLETIZING-[pattern]_i
     B, BX, E, EX ─┘    └── Palletizing number (1 to 16)
```

Fig. 4.4.1 Palletizing instruction

Palletizing is divided into four palletizing patterns according to the stacking and path patterns.

Palletizing patterns

For pallet instructions, palletizing is divided into the following palletizing patterns.
For detail of palletizing patterns, please refer to “10 PALLETIZING FUNCTION”.

Table 4.4.1 Palletizing patterns

Pattern	Allowable palletizing
B	Simple stacking pattern and single path pattern
BX	Simple stacking pattern and multiple path patterns
E	Complex stacking pattern and single path pattern
EX	Complex stacking pattern and multiple path patterns

4.4.2 Palletizing Motion Instruction

The palletizing motion instruction is a motion instruction that uses three path points - an approach point, stack point, and retraction point - as position data. This instruction is dedicated to palletizing. Each palletizing instruction rewrites such position data.

```
            J   PAL_i   [Path point]   100%   FINE

Palletizing numebr                Type of path point
    (1 to 16)                     A_n   :Approach Point    n=1 to 8
                                  BTM   :Stack Point
                                  R_n   :Retract Point     n=1 to 8
```

Fig. 4.4.2 Palletizing motion instruction

4.4.3 Palletizing End Instruction

The palletizing end instruction increments or decrements the value of the palletizing register.

```
        PALLETIZING-END_i
                        Palletizing Number (1 to 16)
```

Fig. 4.4.3 Palletizing end instruction

Example

```
1:     PALLETIZING-B_3
2: L   PAL_3[ A_1 ] 100mm/sec CNT10
3: L   PAL_3[ BTM ] 50mm/sec FINE
4:     HAND1 OPEN
5: L   PAL_3[ R_1 ] 100mm/sec CNT10
6:     PALLETIZING-END_3
```

4.5 REGISTER INSTRUCTIONS

The register instructions perform arithmetic operations on registers. The following register instructions are available:

```
   Instruction  1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Palletizing
8 --next page--
```

- Register instructions
- Position register instructions
- Position register axis instructions
- Palletizing register instructions
- String register, string instructions

In register operations, polynomial operations such as those shown below are possible:

Example 1: R[2]=R[3]-R[4]+R[5]-R[6]
2: R[10]=R[2]*100/R[6]

The following restrictions are imposed:

- Up to five operators can be written on a single line.

Example 1: R[2]=R[3]+R[4]+R[5]+R[6]+R[7]+R[8]
Up to five operators

- The "+" and "-" operators can be mixed on a single line. So can the "*" and "/" operators. "+" and "-" cannot, however, be mixed with "*" and "/".

4.5.1 Register Instructions

A register instruction performs an arithmetic operation on registers. A register is a variable for holding an integer or a decimal fraction. (For registers, See Section 7.3.) Two hundred registers are provided.

R[i] = (value)

The instruction, R[i] = (value), loads a value into a specified register.

```
R[ i ] = (value)
Register number (1 to 200)
  — AR[ i ]
  — Constatnt
  — 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
  — DI[ i ]: Digital input signal
  — DO[ i ] : Digital output signal
  — RI[ i ] : Robot input signal
  — RO[ i ] : Robot 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 cleaned with the timer[i]=RESET instruction.
```

Fig. 4.5.1 (a) Instruction R[i] = (value)

Example
```
1: R[1] = RI[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

Register number (1 to 200)

R[i] =(value) (operator) (vallue) (operator)…

- +
- -
- *
- /
- MOD
- DIV

- AR[i]
- Constatnt
- R[i] : Value of register[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
- DI[i] : Digital input signal
- DO[i] : Digital output signal
- RI[i] : Robot input signal
- RO[i] : Robot 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 cleaned with the timer[i]=RESET instruction.

Fig. 4.5.1 (b) Arithmetic register instruction

Example
```
3: R[3:flag] = DI[4]+PR[ 1, 2 ]
4: R[ R[4] ] = R[1]+1
```

4.5.2 Position Register Instructions

A position register instruction performs an arithmetic operation on position registers. A position register instruction can load position data, the sum of two values, or the difference of two values, into a specified position register. A position register instruction uses the same format as a register instruction.

A position register is a variable for holding position data (x, y, z, w, p, r). (For position registers, see Section 7.4.) One hundred position registers are provided.

> **NOTE**
> Before using position register instructions, lock position registers by specifying "LOCK PREG". When position register instructions are used with the position registers unlocked, operation may become tight. For the "LOCK PREG" instruction, see Section 9.4, POSITION REGISTER LOOK-AHEAD EXECUTION FUNCTION.

PR[i] = (value)

The instruction, PR[i] = (value), loads position data into a specified position register.

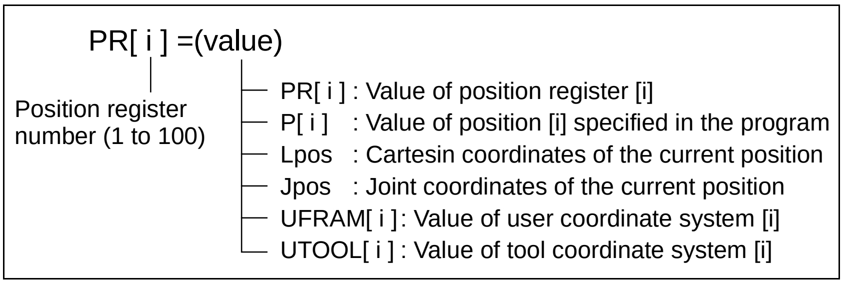


Fig. 4.5.2 (a) Instruction PR[i] = (value)

Example
```
1: PR[1] = Lpos
2: PR[ R[4] ] = 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.

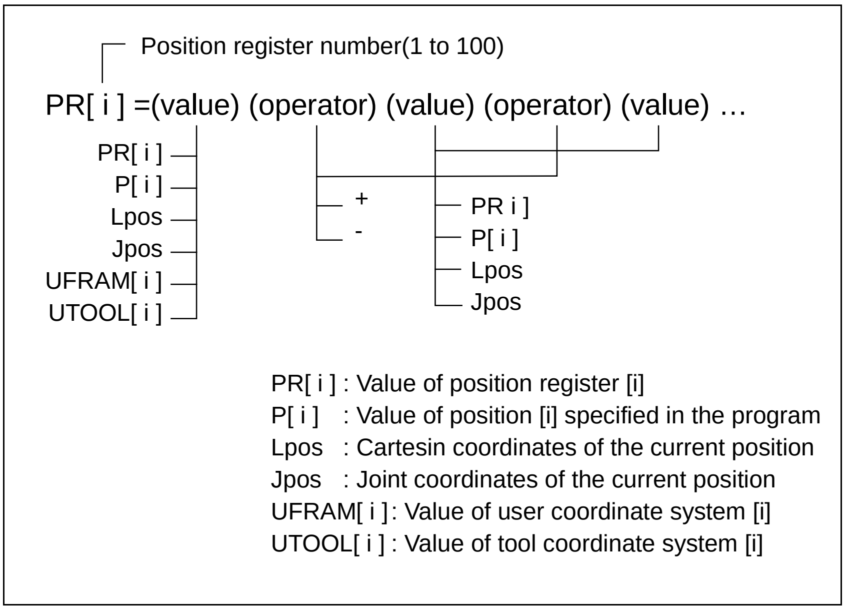


Fig. 4.5.2 (b) PR[i] arithmetic instruction

Example
```
4: PR[3] = PR[3]+Lpos
5: PR[4] = PR[ R[1] ]
```

4.5.3 Position Register Axis Instructions

A position register axis instruction performs an arithmetic operation on position register elements. i of PR[i,j] represents a position register number, and j of PR[i,j] represents a position register element number.

The position register axis instructions can load the value of one position data element, or the sum, difference, product, or quotient of two values into a specified position register element. A position register axis instruction uses the same format as a register instruction.

```
PR[ i , j ]
Position register                    Position register element number
number (1 to 100)
                     Cartesian coordinate system:   Joint coordinate system:
                          1 = X                          1 = J1
                          2 = Y                          2 = J2
                          3 = Z                          3 = J3
                          4 = W                          4 = J4
                          5 = P                          5 = J5
                          6 = R                          6 = J6
                                                         n = Jn
```

Fig. 4.5.3 (a) Format of PR[i,j]

PR[i,j] = (value)

The instruction, PR[i,j] = (value), loads the value of a position data element into a position register element.

```
PR[ i , j ] = (value)
Position register
number (1 to 100)
            AR[ i ]
            Constatnt
            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
            DI[ i ]: Digital input signal
            DO[ i ] : Digital output signal
            RI[ i ] : Robot input signal
            RO[ i ] : Robot 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 cleaned with the timer[i]=RESET instruction.
```

Fig. 4.5.3 (b) Instruction PR[i,j] = (value)

Example
```
1: PR[ 1, 2 ] = R[3]
2: PR[ 4, 3 ] = 324.5
```

PR[I,j] = (value) + (value)

The instruction, PR[i,j] = (value) + (value), loads the sum of two values into a specified position register element.

PR[i,j] = (value) - (value)

The instruction, PR[i,j] = (value) - (value), loads the difference of two values into a specified position register element.

PR[i,j] = (value) * (value)

The instruction, PR[i,j] = (value) * (value), loads the product of two values into a specified position register element.

PR[i,j] = (value) / (value)

The instruction, PR[i,j] = (value) / (value), loads the quotient of two values into a specified position register element.

PR[i,j] = (value) MOD (value)

The instruction, PR[i,j] = (value) MOD (value), loads the remainder (value after decimal point) of the quotient of two values into a specified position register element.

PR[i,j] = (value) DIV (value)

The instruction, PR[i,j] = (value) DIV (value), loads the integer of the quotient of two values into a specified position register element.
R [i] = (x - (x MOD y)) / y

Position register number (1 to 100)

PR[i , j] = (value) (operator) (value) (operator)

Operator:
- +
- -
- *
- /
- MOD
- DIV

Value:
- AR[i]
- Constatnt
- R[i] : Value of register[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
- DI[i] : Digital input signal
- DO[i] : Digital output signal
- RI[i] : Robot input signal
- RO[i] : Robot 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 cleaned with the timer[i]=RESET instruction.

Fig. 4.5.3 (c) PR[i,j] arithmetic instruction

Example
```
3: PR[ 3, 5 ] = R[3]+DI[4]
4: PR[ 4, 3 ] = PR[ 1, 3 ]-3.528
```

4.5.4 Position Register Instructions in Non-Motion Group Program

Position registers and position registers axis instructions are not operated to the group not set group mask. For example, in the system having 3 groups, if group mask is set only first group, position register instructions are operated to only first group.

```
Program detail

  4 Group mask: [1, *, *, *, *, *, *, *]
```

```
PROGRAM
                                           1/4
      1: PR[1] = JPOS
      2: PR[2,4] = R[1]
      3: PR[GP3:3,5] = R[2]
     [End]

1) 2) [ INST ]       3)     4)    5) [EDCMD]  6) >
```

For example, when executed above program, the results of position register operations are followings.
Line 1: The values of JPOS are set to only first group of PR[1]
Line 2: The value of R[1] is set to only first group of PR[2,4]
Line 3: The result is error because the group mask of the third group is not set.

And in non-motion group program, position register operations can’t be executed.

Position Register Instructions in Non-Motion Group Program

If “No motion PR operate mode” is set TRUE on the system configuration screen, position registers operations can be executed in non-motion program. In this case, the results of position register operations has the same result as specifying all the groups that the system has.

Example Three-group system
If position registers operation be executed in non-motion group program, the results of position register operations has the same result as the program (motion group [1, 1, 1, *, *, *, *, *]) in which all motion groups (G1, G2, G3) are set.

```
Program detail

  4 Group mask: [*, *, *, *, *, *, *, *]
```

```
PROGRAM
                                           1/4
      1: PR[1] = JPOS
      2: PR[2,4] = R[1]
      3: PR[GP3:3,5] = R[2]
     [End]

7 8) [ INST ] 9 1 1 12)      [EDCMD]  13)       >
```

“No motion PR operate mode” is set TRUE and executed above program. The results of position register operations are followings.
Line 1: The values of JPOS are set to PR[1] of all groups in the system.
Line 2: The value of R[1] are set to PR[2,4] of all groups in the system.
Line 3: The value of R[2] are set to PR[3,5] of group 3.

Position registers operation in non-motion group program allow user to execute position register operations in parallel with main program execution by multi task function. Please refer to the “9.13 MULTI TASKING FUNCTION” for the details of multi task function.
Using this function, position register operations can be executed without stopping robots motion executed in main task program.

TASK 1

```
Group mask: [1, 1, 1, *, *, *, *, *]

PROGRAM1
                                      4/8
     1: J  P[1] 100% FINE
     2: J  P[2] 70% CNT50
     3: RUN PROGRAM2
     4: L  P[4] 500mm/sec FINE
     5: LOCK PREG
     6: J  PR[1] 100% FINE
     7: UNLOCK PREG
  [End]
```

TASK 2

```
Group mask: [*, *, *, *, *, *, *, *]

PROGRAM2
                                      2/4
     1: PR[1] = JPOS
     2: PR[2,4] = R[1]
     3: PR[GP3:3,5] = R2]
  [End]
```

LOCK PREG instruction

When you execute position register operations by multi task, since the processes of Task 1 and Task 2 proceed at the same time, the value of the position register in use in Task 1 may be rewritten by Task 2.
When use “LOCK PREG” instruction, it is possible to prohibit rewriting the value of the position register from all tasks. Refer to the Section “9.4 POSITION REGISTER LOOK-AHEAD EXECUTION FUNCTION” about “LOCK PREG” instruction.

In the above program, the calculation of position register [1] is performed on the first line of Task 2, but the value of position register [1] is used for the target point on the sixth line of Task 1.
In this case, please prohibit rewriting the value of the position registers from Task 2 by the LOCK PREG before use position register on TASK 1.

> ⚠ **WARNING**
> When you execute position register operations by multi task, you have to use LOCK PREG instruction not to over write the value of the position register used as the position of current motion line. If the position register is over written, it may cause serious effect such as robot moves to unexpected position.

> **NOTE**
> Prohibit rewriting the value of the position registers by all TASK during LOCK PREG instruction. If try to rewrite the value of position register during LOCK PREG instruction, an error occurs and the program pauses. In order to avoid this, we recommend that you create a program that you can check with I/O that calculation of position register has completed.

4.5.5 Arithmetic Palletizing Register Instructions

An arithmetic palletizing register instruction performs an arithmetic operation on palletizing registers. The arithmetic palletizing register instructions are load data, the sum of two values, or the difference between two values. An arithmetic palletizing register instruction uses the same format as a register instruction.
A palletizing register holds palletizing register elements (i, j, k). A program can use up to 32 palletizing registers. (See Section 7.5 PALLETIZING REGISTERS.)

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.

```
[  i , j , k  ]
  └─ Palletizing register elements
      Direct specificastion    : Number of rows, columns, and layers
                                 (1 to 127)
      Indirect specificartion  : Value of register [i]
      Omited                   : Asterisk (*) to specify that no
                                 modification is be made.
```

Fig. 4.5.5 (a) Format of palletizing register elements

PL[i] = (value)

The instruction, PL[i] = (value), loads (assigns) palletizing register elements into a specified palletizing register.

```
                     PL[ i ] =(value)
Palletizing register ─┘      ├─ PL[ i ]     : Palletizing register [ i ]
number (1 to 32)             └─ [ i , j , k ] : Palletizing register elements
```

Fig. 4.5.5 (b) Instruction PL [i] = (value)

Example
```
1: PL[1] = PL[3]
2: PL[2] = [ 1, 2, 1 ]
3: PL[ R[3] ]= [ *, R[1], 1 ]
```

PL[i] = (value) (operator) (value)

The instruction, PL[i] = (value) (operator) (value), performs an arithmetic operation, then loads the result of the operation into a specified palletizing register.

```
     ┌─ Palletizing register number (1 to 32)
PL[ i ] = (value) (operator) (value) (operator)…
├─ +        ├─ PL[ i ]       : Palletizing register [ i ]
└─ -        └─ [ i , j , k ] : Palletizing register elements
```

Fig. 4.5.5 (c) Instruction PL [i] = (value) (operator) (value)

Example
```
1: PL[1] = PL[3]+[ 1, 2, 1 ]
2: PL[2] = [ 1, 2, 1 ]+[ 1, R[1], * ]
```

4.5.6 String Register, String Instructions

A string register stores alphanumeric character strings. Each string register can hold a maximum of 254 characters. The default number of string registers is 25. You can increase the number of string register at controlled start.

SR[i] = (value)

The instruction, SR[i]=(value), loads a value into a specified string register.
Conversion from either integer or floating point numeric data is allowed. Floating point data will be rounded to 6 decimal places.
Conversion from strings to numeric data is allowed. Conversion from string to numeric data will stop when the first alpha character is found.

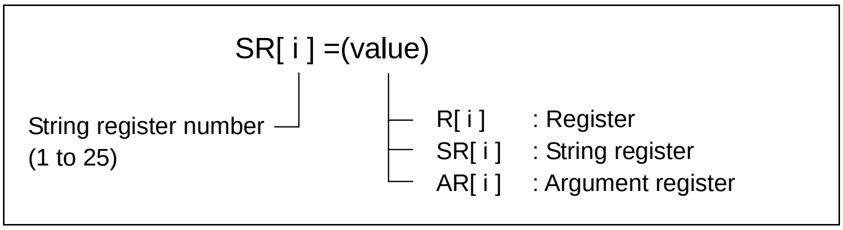


Fig. 4.5.6 (a) Instruction SR[i]=(value)

Example: SR[i] = R[j]

R[j] Value	SR[i] Result
R[j]=1234	SR[i]='1234'
R[j]=12.34	SR[i]='12.34'
R[j]=5.123456789	SR[i]='5.123457'

Example: R[i] = SR[j]

SR[j] Value	R[i] Result
SR[j]='1234'	R[i]=1234
SR[j]='12.34'	R[i]=12.34
SR[j]='765abc'	R[i]=765
SR[j]='abc'	R[i]=0

SR[i] = (value) (operator) (value)

The instruction, SR[i] = (value) (operator) (value), concatenate two values and loads the result into a specified string register.
The data is converted to the data type of the left instruction for each operation.
When the data type of the left instruction is string data, concatenation will be performed.
When the data type of the left instruction is numeric data, arithmetic operation will be performed. Then, if the data type of the right instruction is string data, conversion will stop when the first alpha character is found.

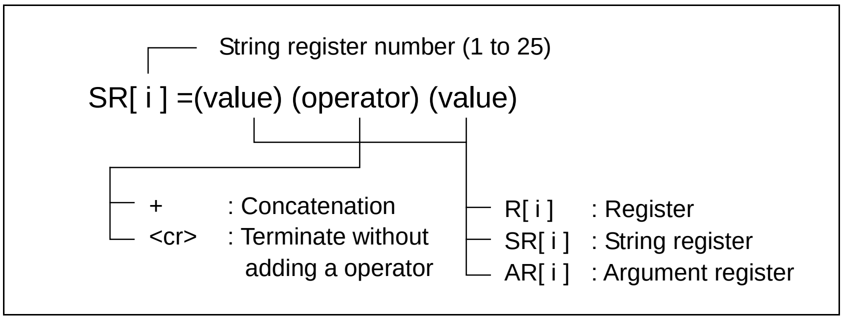


Fig. 4.5.6 (b) Instruction SR[i] = (value) (operator) (value)

Example: SR[i] = R[j] + SR[k]

R[j], SR[k] Value	SR[i] Result
R[j]=123.456+SR[k]='345.678'	SR[i]='469.134'
R[j]=456+SR[k]='1abc2'	SR[i]='457'

Example: SR[i] = SR[j] + R[k]

SR[j], R[k] Value	SR[i] Result
SR[j]=’123.’+R[k]=456	SR[i]=’123.456’
SR[j]=’abc’+R[k]=81573	SR[i]=’abc81573’

Result of consolidated strings, if you exceed 254 characters as a result of consolidated strings, “INTP-323 Value over flow” error occur .

R[i] = STRLEN (value)

The instruction, R[i] = STRLEN (value), returns the length of string stored in the specified register.

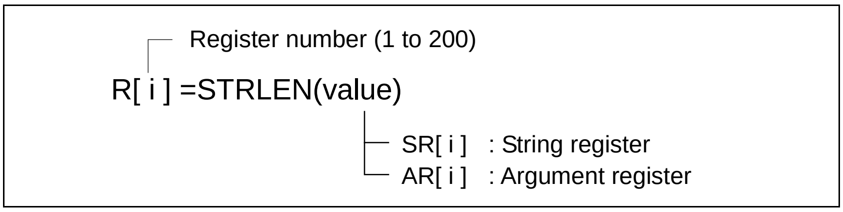


Fig. 4.5.6 (c) Instruction R[i] = STRLEN (value)

Example: R[i] = STRLEN SR[j]

SR[j] Value	R[i] Result
SR[j]=’abcdefghij’	R[i]=10
SR[j]=’abc1,2,3,4,5,6,de’	R[i]=17
SR[j]=’’	R[i]=0

R[i] = FINDSTR (value) (value)

The first (value) means “the target string”, and the second (value) means “the search string”.
The instruction, R[i] = FINDSTR (value) (value), searches the target string and returns the index within the target string. All string comparisons are case insensitive. If the search string is not found within the target string an index of “0” is returned.

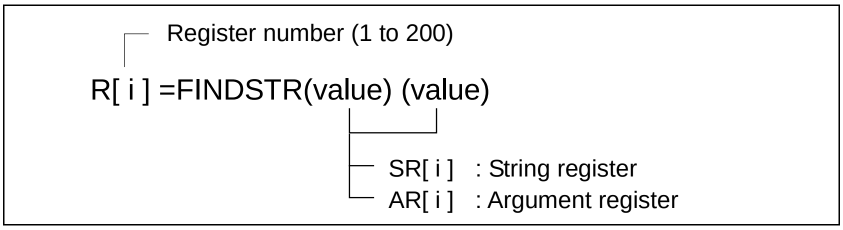


Fig. 4.5.6 (d) Instruction R[i] = FINDSTR (value) (value)

Example: R[i] = FINDSTR SR[j], SR[k]

SR[k] Value, SR[j]=’find this character’	R[i] Result
SR[k]=’find’	R[i]=1
SR[k]=’character’	R[i]=10
SR[k]=’nothing’	R[i]=0
SR[k]=’’	R[i]=0

SR[i] = SUBSTR (value) (value) (value)

The first (value) means “the target string”, the second (value) means “the start point”, and the third (value) means “the length”.
The instruction, SR[i] = SUBSTR (value) (value) (value), returns a substring from the target string based on the start point and length specified.

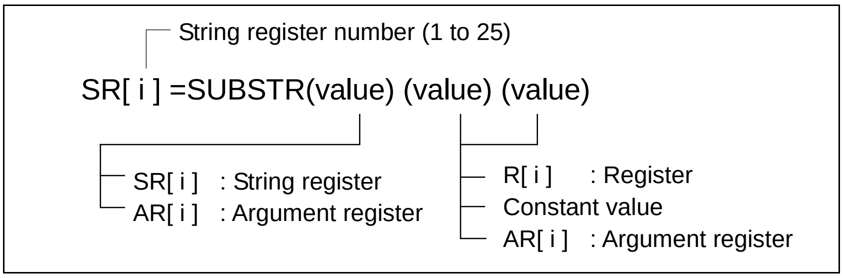


Fig. 4.5.6 (e) Instruction SR[i] = SUBSTR (value) (value) (value)

Example: SR[i] = SUBSTR SR[j], R[k], R[l]

R[k],R[l] Value, SR[j]='This string will be broken apart.'	SR[I] Result
R[k]=1, R[l]=2	SR[i]='This'
R[k]=10, R[l]=7	SR[i]='apart'
R[k]=5, R[l]=0	SR[i]=''

NOTE
Starting point should be larger than "0", and length should be larger than or equal to "0". Starting point plus length should be smaller than the target string length.

4.6 I/O INSTRUCTIONS

The I/O (input/output signal) instructions are used to change the state of a signal output to peripheral devices and read the state of an input signal.

```
   Instruction  1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Palletizing
8 --next page--
```

- (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 needs to be allocated to the physical number before using it. (For configuring I/O, see Section 3.1.)

4.6.1 Digital I/O Instructions

The digital input signal (DI) and digital output signal (DO) are input/output signals that can be controlled by the user.

R[i] = DI[i]

The instruction, R[i] = DI[i] loads, the state of a digital input signal (on = 1/off = 0) into a specified register.

R[i] = DI[i]

Register number (1 to 200) Digital input signal number

Fig. 4.6.1 (a) Instruction R[i] = DI[i]

Example
```
1: R[1] = DI[1]
2: R[ R[3] ] = DI[ R[4] ]
```

DO[i] = ON/OFF

The instruction, DO[i] = ON/OFF, turns on or off a specified digital output signal.

DO[i] = (value)

Digital output signal number

ON : Turns on the digial output signal.
OFF: Turns off the digital output signal.

Fig. 4.6.1 (b) Instruction DO[i] = ON/OFF

Example
```
3: DO[1] = ON
4: DO[ R[3] ] = OFF
```

DO[i] = PULSE,[WIDTH]

The DO[i] = PULSE, [TIME] instruction inverts the current status of a specified digital output for a specified duration. When no duration is specified, pulse output is executed for the duration specified with $DEFPULSE (0.1-second units).

DO[i] = PULSE, (value)

Digital output signal number

Pulse width (sec) (0.1 to 25.5sec)

Fig. 4.6.1 (c) Instructions DO[i] = PULSE,(WIDTH)

Example
```
5: DO[1] = PULSE
6: DO[2] = PULSE, 0.2sec
7: DO[ R[3] ] = PULSE, 1.2sec
```

DO[i] = R[i]

The instruction, DO[i]=R[i], turns on or off a specified digital output signal according to the value of a specified register. When the value of the specified register is 0, the digital output signal is turned off. When the value of the specified register is other than 0, the digital output signal is turned on.

DO[i] = R[i]

Digital output signal number

Register number (1 to 200)

Fig. 4.6.1 (d) Instruction DO[i] = R[i]

Example
```
7: DO[1] = R[2]
8: DO[ R[5] ] = R[ R[1] ]
```

4.6.2 Robot I/O Instructions

The robot input signal (RI) and robot output signal (RO) are input/output signals that can be controlled by the user.

R[i] = RI[i]

The instruction, R[i] = RI[i], loads the state of a robot input signal (on = 1/off = 0) into a specified register.

```
R[ i ] = RI[ i ]
Register number (1 to 200)    Robot input signal number
```

Fig. 4.6.2 (a) Instruction R[i] = RI[i]

Example
```
1: R[1] = RI[1]
2: R[ R[3] ] = RI[ R[4] ]
```

RO[i] = ON/OFF

The instruction, RO[i] = ON/OFF, turns on or off a specified robot output signal.

```
RO[ i ] = (value)
Robot output          ON  : Turns on the robot output signal.
signal number         OFF: Turns off the robot output signal.
```

Fig. 4.6.2 (b) Instruction RO[i] = ON/OFF

Example
```
3: RO[1] = ON
4: RO[ R[3] ] = OFF
```

RO[i] = PULSE,[WIDTH]

The RO[i] = PULSE,[TIME] instruction inverts the current status of a specified robot output for a specified duration. When no duration is specified, pulse output is executed for the duration specified with $DEFPULSE (0.1-second units).

```
RO[ i ] = PULSE, (value)
Robot output signal           Pluse width (sec)
number                        (0.1 to 25.5)
```

Fig. 4.6.2 (c) Instruction of RO[i] = PULSE,[WIDTH]

Example
```
5: RO[1] = PULSE
6: RO[2] = PULSE, 0.2sec
7: RO[ R[3] ] = PULSE, 1.2sec
```

RO[i] = R[i]

The instruction, RO[i] = R[i], turns on or off a specified robot output signal according to the value of a specified register. When the value of the specified register is 0, the robot output signal is turned off. When the value of the specified register is other than 0, the robot output signal is turned on.

```
RO[ i ] = R[ i ]
Robot output signal     Register number (1 to 200)
number
```

Fig. 4.6.2 (d) Instruction RO[i] = R[i]

Example
```
7: RO[1] = R[2]
8: RO[ R[5] ] = R[ R[1] ]
```

4.6.3 Analog I/O Instructions

Analog input (AI) and analog output (AO) signals indicate levels as a value on a continuum. Thus, the magnitude of a signal represents a temperature, voltage, or other data.

R[i] = AI[i]

The R[i] = AI[i] instruction stores the value of an analog input signal in a register.

R[i] = AI[i]

Register number (1 to 200) — Analog input signal number

Fig. 4.6.3 (a) R[i] = AI[i] Instruction

Example
```
1: R[1] = AI[1]
2: R[ R[3] ] = AI[ R[4] ]
```

AO[i] = (value)

The AO[i] = (value) instruction outputs a value as a specified analog output signal.

AO[i] = (value)

Analog output signal number — Value of analog output signal

Fig. 4.6.3 (b) AO[i] = (value) Instruction

Example
```
3: AO[1] = 0
4: AO[ R[3] ] = 3276.7
```

AO[i] = R[i]

The AO[i] = R[i] instruction outputs a register value as an analog output signal.

AO[i] = R[i]

Analog output signal number — Register number (1 to 200)

Fig. 4.6.3 (c) AO[i] = R[i] Instruction

Example
```
5: AO[1] = R[2]
6: AO[ R[5] ] = R[ R[1] ]
```

4.6.4 Group I/O Instruction

R [i] = GI [i]

The signal of the group input(GI) and the group output(GO) is that some digital input/output signals are grouped and this is controlled by one instruction.
The instruction, R[i]=GI[i], converts the binary value of the specified group input signal to the decimal value and inputs it to the specified register.

R[i] = GI[i]

Register number (1 to 200) — Group input signal number

Fig. 4.6.4 (a) Instruction R [i] = GI [i]

Example
```
1: R[1] = GI[1]
2: 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.

GO[i] = (value)

Group output signal number

Value of group output signal

Fig. 4.6.4 (b) Instruction GO [i] = (value)

Example
```
3: GO[1] = 0
4: GO[ R[3] ] = 32767
```

GO [i] = R [i]

The GO[i]=R[i] instruction sends the binary equivalent of the contents of specified register on the specified group output lines.

GO[i] = R[i]

Group output signal number

Register number (1 to 200)

Fig. 4.6.4 (c) Instruction GO [i] = R [i]

Example
```
5: GO[1] = R[2]
6: GO[ R[5] ] = R[ R[1] ]
```

4.7 BRANCH INSTRUCTIONS

A branch instruction causes a branch from one line of a program to another. Four types of branch instructions are supported.

- Label instruction
- Program end instruction
- Unconditional branch instruction
- Conditional branch instruction

4.7.1 Label Instruction

LABEL[i]

The label instruction (LBL[i]) is used to specify a program execution branch destination. A label is defined with a label definition instruction.

```
   Instruction  1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Palletizing
8 --next page--
```

A comment can be added to explain a label. Once a label is defined, it can be used for either an unconditional branch or conditional branch. It is not possible to specify the label number as the indirect addressing. To add a comment, move the cursor to the label number and press the [ENTER] key.

```
LBL[ i : Comment ]
```

Label (1 to 32766)

A comment can consist of up to 16 characters including alphanumeric characters, asterisks(*), underlines(_), and at marks (@), etc.

Fig. 4.7.1 LBL[i] Instruction

Example
```
1: LBL[1]
2: LBL[ R[3] ]
```

4.7.2 Program End Instruction

END

The program end instruction indicates the end of a program. The execution of a program is terminated by this instruction. If a program is called from another main program, control is returned to the main program.

```
   Instruction  1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Palletizing
8 --next page--
```

```
END
```

Fig. 4.7.2 Program end instruction

4.7.3 Unconditional Branch Instructions

An unconditional branch instruction invariably causes a branch from one line to another. 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.

```
JMP   LBL[ i ]
```

Label (1 to 32766)

Fig. 4.7.3 (a) JMP LBL[i] instruction

Example
```
3: JMP LBL[2]
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 menu automatically displayed or press F5, "STRINGS" to enter characters directly.

```
CALL (Program)
        └─ Name of a program to be called
```

Fig. 4.7.3 (b) CALL (program) instruction

Example
```
5: CALL SUB1
6: CALL PROGRAM2
```

*) It is possible to set an argument for the program call instruction and use its value in a subprogram. See Subsection 4.7.6, "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  1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Palletizing
8 --next page--
```

- 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.

```
IF (variable) (operator) (value) (processing)

            R[ i ] ─┐   ├─ >    ├─ Constant  ├─ JMP LBL[ i ]
 $ System variable ─┘   ├─ >=   └─ R[ i ]    └─ CALL (Program)
                        ├─ =
                        ├─ <=
                        ├─ <
                        └─ <>
```

Fig. 4.7.4 (a) Register conditional compare instruction

> ⚠ **CAUTION**
> When the contents of a register is compared with the real value using the operator "=", the contents does not always correspond to the real value because of the rounding-off error of the contents. To compare with the real value, use the operator without 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.

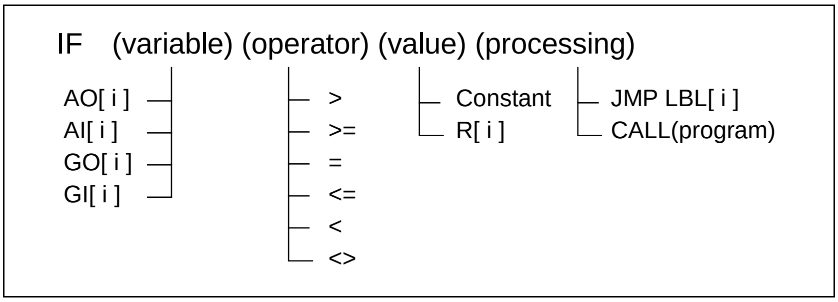


Fig. 4.7.4 (b) I/O conditional compare instruction 1

Example
```
7: IF R[1] = R[2], JMP LBL[1]
8: IF AO[2] >= 3000, CALL SUBPRO1
9: IF GI[ R[2] ] = 100, CALL SUBPRO2
```

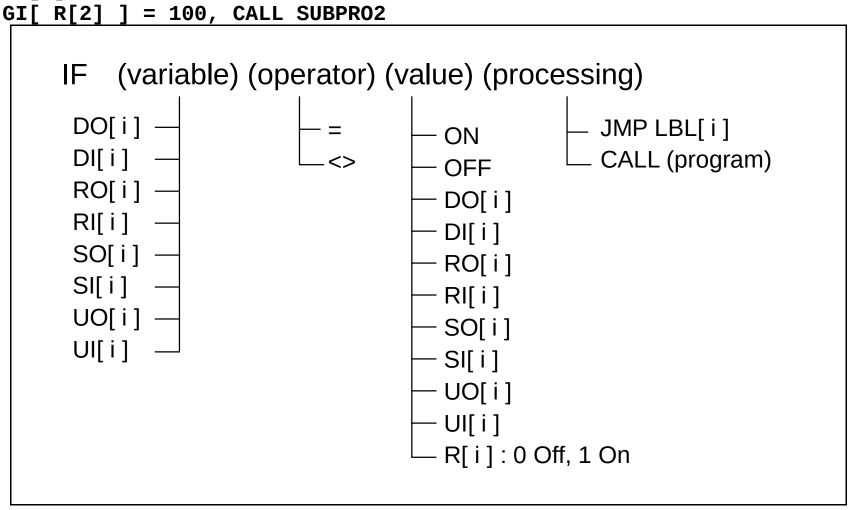


Fig. 4.7.4 (c) I/O conditional compare instruction 2

Example
```
10: IF RO[2] <> OFF, JMP LBL[1]
11: IF DI[3] = ON, CALL SUB
```

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]

Pallet register conditional compare instruction

IF PL [i] (operator) (value) (processing)

The pallet register conditional compare instruction compares the value of the pallet register with the value of another pallet register element. When the comparison is true, specified processing is executed. If 0 is entered in each element, "*" appears. Only a numeric value or remainder specification can be used for each element to be compared.

For the pallet register element, specify the element whose value is to be compared with the value of the pallet register. There are four methods for specification.

[i , j , k]

Palletizing register element

Direct specification : Number of rows, columns, and layers (1 to 127)
Indirect specification : Value of register [i]
Remainder specification : a-b : What results in a remainder of b when divided by a (a : 1 to 127, b : 0 to 127)
No specification : * is any value.

Fig. 4.7.4 (d) Pallet register element format

IF PL[i] (operator) (value) (processing)

Palletizing register number (1 to16)

- (operator): = / <>
- (value): PL[i] / [i , j , k]
- (processing): JMP LBL[i] / CALL (program)

Fig. 4.7.4 (e) Pallet register conditional compare instruction

Example

12: IF PL[1] = R[2], JMP LBL［1］
13: IF PL[2]<>［1, 1, 2］, CALL SUB1
14: IF PL[R[3]]<>［*, *, 2-0］, CALL SUB1

Conditional select instruction

SELECT R[i] = (value) (processing)
= (value) (processing)
= (value) (processing)
ELSE (processing)

The conditional select instruction consists of several register compare instructions. The conditional select instruction compares the value of a register with one or more values, then selects a statement that satisfies the comparison condition.

- If the value of a specified register matches one value, the jump instruction or subprogram call instruction corresponding to the value is executed.
- If the value of a specified register does not match any of the values, the jump instruction or subprogram call instruction corresponding to ELSE is executed.

- If the one condition is satisfied and statement is executed, the other statement is not executed even though the condition is satisfied.

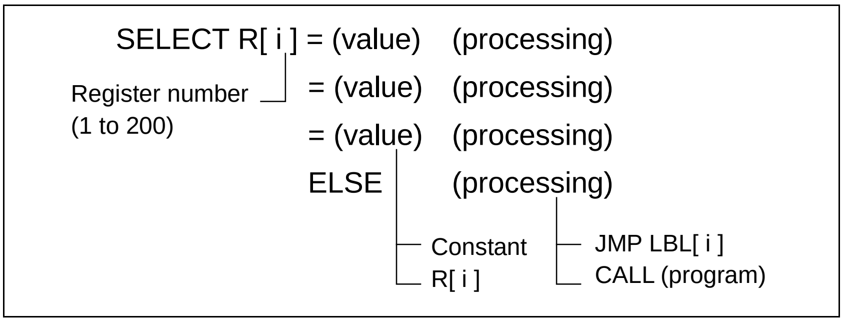


Fig. 4.7.4 (f) Conditional select instruction

Example 1

```
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
```

Example 2

```
10: R[1]=2
11: SELECT R[1]=1, CALL PROG1
12:        = 2, CALL PROG2
13:        = 3, CALL PROG3
14:        = 2, CALL PROG4
15:        ELSE, CALL SUB2
```

In Example 2, R[1] equals to 2, so the conditions are matched in line 12 and line 14. But when one condition is satisfied, another statement does not executed even though condition is satisfied. The PROG4 is not called in line 14, because the PROG2 is called in line 12.

> ⚠ **CAUTION**
> If the one condition is satisfied and statement is executed, another statement is not executed whether the condition is satisfied or not in conditional select instruction.

4.7.5 IF_THEN/ELSE/ENDIF Statement

IF_THEN /ELSE/ENDIF statements are conditional branch instructions.
When the condition is satisfied, program lines enclosed with IF_THEN and ENDIF statements (or ELSE statement) are executed.
When the condition is not satisfied, program lines enclosed ELSE and ENDIF statements are executed.

> **NOTE**
> IF/ELSE/ENDIF statements can be used in the version 7DC3 or later.

IF_THEN statement

IF_THEN statement is used with ENDIF statement in pairs.
When the condition is satisfied, lines below IF_THEN statement are executed.
If ELSE statement is used, lines enclosed with ELSE and ENDIF statements are not executed.

```
IF  (Condition)  THEN
```

Mixed Logic instructions are used for condition.

ELSE statement

ELSE statement is used with IF_THEN and ENDIF statement.
When the condition of IF_THEN statement is not satisfied, lines below ELSE statement are executed.
Lines enclosed with IF_THEN and ELSE statements are not executed.

```
ELSE
```

ENDIF statement

ENDIF statement is used with IF_THEN statement in pares.
This is a statement that indicate end of IF_THEN and ELSE statements.

```
ENDIF
```

IF_THEN, ELSE and ENDIF statements are automatically combined after teaching.
Nested structure can be formed by teaching IF_THEN/ENDIF statements in areas enclosed with IF_THEN/ENDIF or FOR/ENDFOR.
Up to 10 depth of nested structure can be formed. However, teaching more than 10 nested loops causes an alarm in execution.
The number of IF_THEN statement and ENDIF statement in a program should be the same. When the number is not the same, an alarm occurs in execution.

Examples:
If condition of IF_THEN statement on line 3 is satisfied (DI[1]=ON), line 4 and 5 are executed. In this case, execution line jump to ENDIF line when ELSE statement is executed. Line 7 and 8 are not executed.

```
 1: L P[1] 100mm/sec FINE
 2: L P[2] 100mm/sec FINE
 3:  IF (DI[1]=ON) THEN
 4: L P[3] 100mm/sec CNT100
 5: L P[4] 100mm/sec CNT100
 6:  ELSE
 7: L P[5] 100mm/sec CNT100
 8: L P[6] 100mm/sec CNT100
 9:  ENDIF
10: L P[7] 100mm/sec FINE
11: L P[8] 100mm/sec FINE
```

If condition of IF_THEN statement on line 3 is not satisfied (DI[1]=OFF), execution line jump to ELSE line when IF_THEN statement is executed. Line 4 and 5 are not executed. After that, lines below ELSE statements are executed.

```
 1: L P[1] 100mm/sec FINE
 2: L P[2] 100mm/sec FINE
 3:  IF (DI[1]=ON) THEN
 4: L P[3] 100mm/sec CNT100
 5: L P[4] 100mm/sec CNT100
 6:  ELSE
 7: L P[5] 100mm/sec CNT100
 8: L P[6] 100mm/sec CNT100
 9:  ENDIF
10: L P[7] 100mm/sec FINE
11: L P[8] 100mm/sec FINE
```

Procedure 4-3 Teaching of IF_THEN/ELSE/ENDIF statements

Step

1 Push F1 key [INST]. List of instructions are displayed.

```
PROGRAM1                                              .
                                                  1/1
     [END]

          Instruction  1
          1 Registers
          2 I/O
          3 IF/SELECT
          4 WAIT
          5 JMP/LBL
          6 CALL
          7 Miscellaneous
          8 --next page--

   [INST]                             [EDCMD]
```

2 Select ‘IF/SELECT’ item. List of condition branch instructions is displayed.

```
PROGRAM1                                              .
                                                  1/2
        1:
     [END]              IF statement 2
                IF stat 1 IF (…) THEN
                1 IF …= 2 ELSE
                2 IF …< 3 ENDIF
                3 IF …< 4 SELECT R[]=
                4 IF …< 5 <SELECT> =…
                5 IF …> 6 <SELECT>ELSE
                6 IF …> 7
                7 IF (… 8 --next page--
                8 --next page--

   [INST]                             [EDCCMD]
```

3 Select ‘IF (…) THEN’ item. IF_THEN statement is added. And Cursor moves to space for condition and list of mixed logic instructions are displayed.

```
PROGRAM1                                              .
                                                  1/2
        1:  IF (...) THEN
     [END]
             Mixed Logic 1
             1 (
             2 DI[]
             3 DO[]
             4 R[]
             5 F[]
             6 ON
             7 OFF
             8 --Next page--

   \<INSERT>      <DELETE>        [CHOICE]       (!)
```

4 Teach condition expression using mixed logic instructions. Refer to “4.19 MIXED LOGIC INSTRUCTION” to teach mixed logic instructions.

```
PROGRAM1                                              .
                                                  1/2
        1: IF (DI[1]=ON) THEN
     [END]

     <INSERT>
```

5 Teach instructions below the line with IF_THEN statement that is executed when condition is satisfied.

```
PROGRAM1                                              .
                                                   5/5
        1:  IF (DI[1]=ON) THEN
        2: L P[1] 100mm/sec CNT100
        3: L P[2] 100mm/sec CNT100
        4: L P[3] 100mm/sec FINE
     [END]

   <INSERT>
```

6 ELSE is taught if it is needed. Push F1 key [INST] and select ‘IF/SELECT’ item.

```
PROGRAM1                                              .
                                                   5/5
        1:  IF (DI[1]=ON) THEN
        2: L P[1] 100mm/sec CN IF statement 2
        3: L P[2] 100m IF stat 1 IF (…) THEN
        4: L P[3] 100m 1 IF …= 2 ELSE
     [END]             2 IF …< 3 ENDIF
                       3 IF …< 4 SELECT R[]=
                       4 IF …< 5 <SELECT> =…
                       5 IF …> 6 <SELECT>ELSE
                       6 IF …> 7
                       7 IF (… 8 --next page--
                       8 --next page--

  [INST]                                   [EDCCMD]
```

7 Select ‘ELSE’ item. ELSE statement is added.

```
PROGRAM1                                              .
                                                   6/6
        1:  IF (DI[1]=ON) THEN
        2: L P[1] 100mm/sec CNT100
        3: L P[2] 100mm/sec CNT100
        4: L P[3] 100mm/sec FINE
        5:  ELSE
     [END]

  [INST]                                   [EDCCMD]
```

8 Teach instructions below the line with ELSE statement that is executed when condition is not satisfied.

```
PROGRAM1                                              .
                                                 10/10
        1:  IF (DI[1]=ON) THEN
        2: L P[1] 100mm/sec CNT100
        3: L P[2] 100mm/sec CNT100
        4: L P[3] 100mm/sec FINE
        5:  ELSE
        7: L P[4] 100mm/sec CNT100
        8: L P[5] 100mm/sec CNT100
        9: L P[6] 100mm/sec FINE
     [END]

  [INST]                                   [EDCCMD]
```

9 Push F1 key [INST] and select ‘IF/SELECT’ item.

```
PROGRAM1                                              .
                                                 10/10
        1:  IF (DI[1]=ON) THEN
        2: L P[1] 100mm/sec CN IF statement 2
        3: L P[2] 100mm IF stat 1 IF (…) THEN
        4: L P[3] 100mm 1 IF …= 2 ELSE
        5:  ELSE        2 IF …< 3 ENDIF
        7: L P[4] 100mm 3 IF …< 4 SELECT R[]=
        8: L P[5] 100mm 4 IF …< 5 <SELECT> =…
        9: L P[6] 100mm 5 IF …> 6 <SELECT>ELSE
     [END]              6 IF …> 7
                        7 IF (… 8 --next page--
                        8 --next page--

  [INST]                                   [EDCCMD]
```

10 Select ‘ENDIF’ item. ENDIF statement is added.

```
PROGRAM1                                              .
                                              11/11
        1:  IF (DI[1]=ON) THEN
        2: L P[1] 100mm/sec CNT100
        3: L P[2] 100mm/sec CNT100
        4: L P[3] 100mm/sec FINE
        5:  ELSE
        7: L P[4] 100mm/sec CNT100
        8: L P[5] 100mm/sec CNT100
        9: L P[6] 100mm/sec FINE
       10:  ENDIF
      [END]

   [INST]                          [EDCCMD]
```

11 When cursor is on THEN, ELSE or ENDIF, line number of paired statements are displayed in prompt line. It helps to check paired statements and modify programs.
Error message is displayed when number of IF_THEN, ELSE and ENDIF statements are not matched.

```
PROGRAM1                                              .
                                               1/11
        1:  IF (DI[1]=ON) THEN
        2: L P[1] 100mm/sec CNT100
        3: L P[2] 100mm/sec CNT100
        4: L P[3] 100mm/sec FINE
        5:  ELSE
        7: L P[4] 100mm/sec CNT100
        8: L P[5] 100mm/sec CNT100
        9: L P[6] 100mm/sec FINE
       10:  ENDIF
      [END]

        <Line Num.>  ELSE:5, ENDIF:10

   [INST]                          [EDCCMD]
```

```
PROGRAM1                                              .
                                               1/10
        1:  IF (DI[1]=ON) THEN
        2: L P[1] 100mm/sec CNT100
        3: L P[2] 100mm/sec CNT100
        4: L P[3] 100mm/sec FINE
        5:  ELSE
        7: L P[4] 100mm/sec CNT100
        8: L P[5] 100mm/sec CNT100
        9:
      [END]

        Need ENDIF for IF in line

   [INST]                          [EDCCMD]
```

Other examples of execution

Backward execution

Backward execution is performed on IF_THEN statement. It is also performed in Area enclosed with IF_THEN and ENDIF statements.
Backward execution is prohibited on ELSE and ENDIF statements. The alarm “INTP-238 BWD execution completed” occurs when backward is executed on these statements.

Start programs in lines enclosed with IF_THEN and ENDIF statements

Program can be started from lines enclosed with IF_THEN and ENDIF statements.
After program is started, execution line jumps to line of ENDIF statement when ELSE statement is executed.
And when program is started on line of ELSE statement, execution line jumps to line of ENDIF statements.

```
      :  IF (DI[1]=ON) THEN
Start : L P[1] 100mm/sec CNT100
      : L P[2] 100mm/sec CNT100
      : L P[3] 100mm/sec FILE
      :  ELSE
      : L P[4] 100mm/sec CNT100
      : L P[5] 100mm/sec CNT100
      : L P[6] 100mm/sec FINE
      :  ENDIF
      : L P[7] 100mm/sec FINE
      : L P[8] 100mm/sec CNT100
```

JMP/LBL statement in area enclosed with IF_THEN and ENDIF statements

JMP/LBL statements can be used in area enclosed with IF_THEN and ENDIF statements. In following program, cursor jump to area enclosed with IF_THEN and ENDIF by JMP statement. When ELSE statements is executed even if cursor jump to IF_THEN area by JMP statement, execution line jump to line of ENDIF.

```
:  JMP LBL[1]
:  IF (DI[1]=ON) THEN
: L P[1] 100mm/sec CNT100
:  LBL[1]
: L P[2] 100mm/sec FINE
:  ELSE
: L P[2] 100mm/sec CNT100
: L P[3] 100mm/sec CNT100
: L P[5] 100mm/sec FINE
:  ENDIF
: L P[6] 100mm/sec FINE
: L P[7] 100mm/sec CNT100
```

> **NOTE**
> Please be careful when you use JMP/LBL in area enclosed with IF_THEN and ENDIF statements.
> Before you use them, please consider how it works.

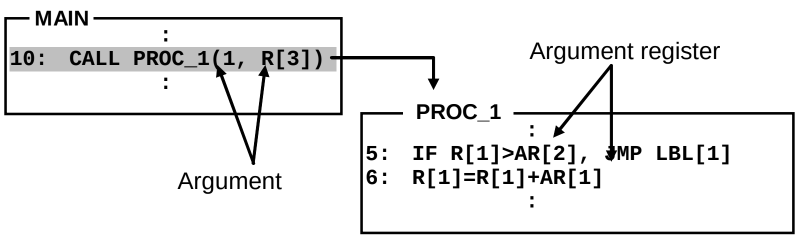
Alarms

- When IF_THEN statements is less numerous than ENDIF statements, INTP-690 (program, line) Need IF_THEN occurs.
- When ENDIF statements is less numerous than IF_THEN statements INTP-691 (program, line) Need ENDIF occurs.
- When depth of nested structure by FOR and IF_THEN statements is over 10, INTP-694 (program, line) Too many FOR/IF_THEN occurs.
- When IF_THEN, ELSE or ENDIF statement is enclosed with FOR and ENDFOR statements, INTP-690 (program, line) Need IF_THEN or INTP-693 (program, line) Need ENDFOR occurs.

4.7.6 Arguments

By using ”arguments” and ”argument registers,” it is possible to transfer data between two programs only.

Example)
In this example, the main program "MAIN" calls the subprogram "PROC_1" with two arguments. "PROC_1" can use the values of the arguments with the argument registers. The first argument corresponds to "AR[1]" while the second argument corresponds to "AR[2]".



Arguments can be used in macro instructions in the same way.

Argument types

The following arguments are supported.

Table 4.7.6 (a) Argument types

Argument type	Example
Constant	1, 3.5
Character string	’Perch’
Argument register*	AR[3]
Register	R [6]
String register	SR[4]

* Argument is used as argument register in subprograms.

```
CALL PROC_1        (1,     AR[1],  R[6])
                    ↓       ↓       ↓
PROC_1 :         AR[1],  AR[2],  AR[3]
```

Instructions for which arguments can be set

Table 4.7.6 (b) Instructions for which arguments can be set

Instruction	Example
Program call instruction	CALL SUBPRG_1 (1, R[3], AR[1])
Macro instruction	Vacuum_hand_open (2.5)

> ⚠ **CAUTION**
> A program call used for branching with an instruction such as a conditional branch instruction cannot use arguments. This problem can be solved by programming as follows:
> (Arguments cannot be set) (Arguments can be set)
> IF R[1]=3, CALL PROC_5 → IF R[1]<>3, JMP LBL[1]
> CALL PROC_5 (1, R[2])
> LBL[1]

Instructions that can use argument registers

Table 4.7.6 (c) Instructions that can use argument registers

Instruction	Example
Right side of an instruction and conditional expression having a register on the left side	R[1]=AR+R[2]+AR[4] 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]

Instruction	Example
Right side of a conditional expression having analog input/output (AI[]/AO[]) or group input/output (GI[]/GO[]) on the left side	IF AO[7]=AR[1], JMP LBL[1] WAIT GI[1]<>AR[2], TIMEOUT, LBL[1]
Right side of the user coordinate system selection instruction and the tool coordinate system selection instruction	UTOOL_NUM=AR[4]
Indirect index specification	R[AR[1]]=R[AR[2]] DO[AR[1]]=ON
Argument of a program call instruction	CALL SUBPRG1 (AR[5])
Argument of a macro instruction	hand 3 open (AR[1])

Restrictions on arguments

The following restrictions are imposed on arguments:

- Up to 30 arguments can be set.
- An argument of character string type can be one to 34 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 menu, press function key [CHOICE].

```
Parameter select 1
1 R[ ]
2 Constant
3 String
4 AR[ ]
5 <None>
6 <Insert>
7 SR[ ]
8
```

Specifying arguments of the constant type

To specify an argument of the constant type, press the [CHOICE] function key and select ”2. Constant” from the menu (see ”Specifying arguments”).

```
Parameter select  1
1 R[ ]
2 Constant
3 string
4 AR[ ]
5 <None>
6 <Insert>
7 SR[ ]
8
```

```
1:  CALL PROC_1 (Constant)
```

Enter the value.

```
1:  CALL PROC_1 (1)      Enter “1”.
```

Specifying arguments of character type

To specify an argument of character type, press function key [CHOICE] and select "3. String" from the menu (see ”Specifying arguments”). The character string type selection menu will be displayed.

```
MAIN
                                              1/2
        1:   CALL PROC_1(1, '    ')
     [End]
   String select 1
   1 PARTS
   2 TOOL
   3 WORK
   4 POS
   5 DEV
   6 PALT
   7 GRIP
   8 --next page--

                                  STRINGS
```

When a character string type is selected, the character string selection menu will be displayed.

```
 String select  1
1 Parts_ITEM1
2 Parts_ITEM2
3 Parts_ITEM3
4 Parts_ITEM4
5 Parts_ITEM5
6 Parts_ITEM6
7 Parts_ITEM7
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.

```
MAIN
                                             1/2
        1:   CALL PROC_1('TOOL12       ')
     [End]





                                    Alpha input 1
                                    1 Words
                                    2 Upper Case
                                    3 Lower Case
                                    4 Options
      $         [         ]
```

Press the [ENTER] key to decide the character string.

1: CALL PROC_1 (‘TOOL12’)   “TOOL12” is decided.

To change a character string, move the cursor to the character string and press F5, "CHANGE". The character string type selection menu will be displayed .

Specifying arguments of the argument register type

To set an argument of the argument register type, press the [CHOICE] function key and select "4 AR[]" from the menu (see ”Specifying arguments").

```
Parameter select 1
1 R[ ]
2 Constant
3 String
4 AR[ ]
5 <None>
6 <Insert>
7 SR[ ]
8
```

1: CALL PROC_1 (AR[...])

Enter the index.

1: CALL PROC_1 (AR[1])   Enter "1" to the index.

To toggle between direct and indirect index specifications, press F3, "INDIRECT". 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 F4, [CHOICE] and select ”1 R[ ]” from the submenu (see ”Specifying arguments").

```
Parameter select1
1 R[ ]
2 Constant
3 String
4 AR[ ]
5 <None>
6 <Insert>
7 SR[ ]
8
```

1: CALL PROC_1 (R[...])

Enter the index.

1: CALL PROC_1 (R[1])   Enter "1" to the index.

To toggle between direct and indirect index specifications, press the F3, "INDIRECT". 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 menu (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.

1: CALL PROC_1 (1, 2)

Press function key [CHOICE] and select "6 <Insert>" from the menu (see ”Specifying arguments). A new argument can be inserted at the cursor position.

```
Parameter select 1
1 R[ ]
2 Constant
3 String
4 AR[ ]
5 <None>
6 <Insert>
7 SR[ ]
8
```

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 menu reappears; select the argument type.

Deleting arguments

Position the cursor to the argument to be deleted.

1: CALL PROC_1 (1, 2 , 3)

Press function key [CHOICE] and select "5 <None>" from the menu (see ”Specifying arguments”). The argument is deleted from the cursor position.

```
Parameter select 1
1 R[ ]
2 Constant
3 String
4 AR[ ]
5 <None>
6 <Insert>
7 SR[ ]
8
```

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 menu; 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:

```
REGISTER statement 1
1 R[ ]
2 Constant
3 DO[ ]
4 DI[ ]
5 RO[ ]
6 RI[ ]
7 GO[ ]
8 --next page--
```

```
REGISTER statement 2
1 GI[ ]
2 SO[ ]
3 SI[ ]
4 UO[ ]
5 UI[ ]
6 TIMER
7 TIMER_OVERFLOW
8 --next page--
```

```
REGISTER statement 3
1 AR[ ]
2 AO[ ]
3 AI[ ]
4 PR[i,j]
5 SR[ ]
6
7
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]

If F3, "INDIRECT" is pressed twice at an element having an index, an argument register can be used for an indirect index specification.

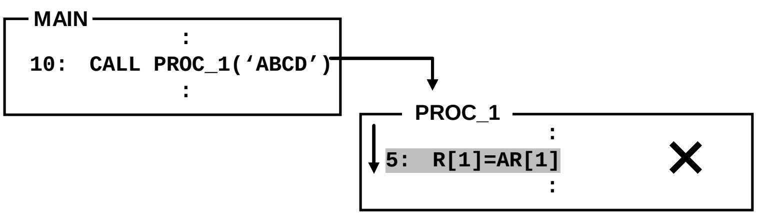
1: WAIT R[R ...]]	When F3 is pressed once
1: WAIT R[AR ...]]	When F3 is pressed again

Notes on using arguments

Note the following when specifying arguments:

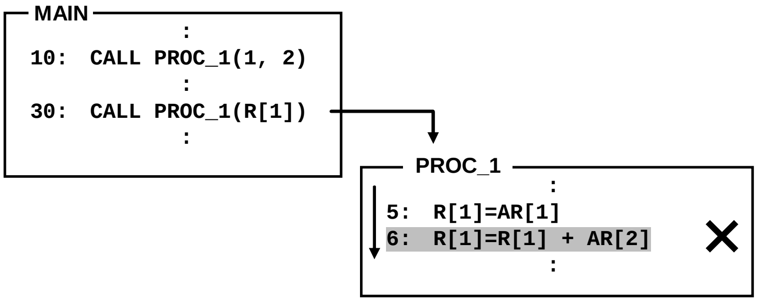
- The contents of an argument are not checked when the argument is specified. If the type of an argument does not match the type of the corresponding one in the subprogram, an error occurs during execution.

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 4.7.6 (d) System variables relating to arguments

Item	System variable	Remarks
1 Use of character string	$STRING_PRM=TRUE/FALSE Standard value=FALSE	
2 Character string type	$ARG_STRING[i].$TITLE (i = 1 to 10)	More than 1 and up to 16 characters
3 Character string	$ARG_STRING[i].$ITEMj (i = 1 to 10, j=1 to 20)	Up to 16 characters
4 "Word" at character entry	$ARG_WORD[i] (i = 1 to 5)	Up to 7 characters

4.7.7 Wizard to Input Arguments

4.7.7.1 Overview

Wizard to input arguments allows you to use a wizard to teach call instruction arguments in any sequence. For example, there is a program called, TRACKING.

```
1: CALL TRACKING('CStn_Out_R1',2,(-1),1,1,1)
```

This program needs to be called with six arguments of a specific type and in the correct order.

Using Wizard to input arguments, allows you to teach the arguments in any sequence. The argument description is also displayed.

```
1: CALL TRACKING(Area Name='CStn_Out_R1',VR num=2,
 : Timeout Time=(-1),Reg num timeout=1,NOT-CONSECUTIVE,
 : Model ID=1)
```

By using this function, you do not need to memorize the number, type, order of arguments, so you can easily teach arguments.

4.7.7.2 Preparation

To use this function, you must prepare and load a text file (DT file) that includes information for the wizard.

File Format

The following is an example text (.DT) file.

```
[PROGRAM]      {Sample format}
NAME = "TRACKING"
ARGUMENT = '6'
[ARGUMENT]
S01 = "Area Name"
N02 = "VR num"
N03 = "Timeout Time"
N04 = "Reg num timeout"
V05 = "NOT-CONSECUTIVE":'1',
      "CONSECUTIVE":'2'
N06 = "Model ID"
[ENDPROGRAM]
```

```
< Program information >
NAME     : Called program name
ARGUMENT : Number of argument
```

```
< Argument type >
N : Constant number
V : Selectable macro definitions
S : String
W : Selectable prepared strings
```

How to Write the Text File

The text file details are described below.

1. [PROGRAM] and under
   - NAME is for the program name to call with arguments. (Max length is 36 characters)
   - ARGUMENT is the number of the argument. (Max. is 30.)

2. [ARGUMENT] and under
   - Basic writing style is | N(type of argument) 02(what number for argument) =(equal mark) "VR num"(displayed meaning of argument)|
     - The number of (type of argument) is 4 types (N: Constant number, V: Macro definition, S: String, W: List of string)
     - (What number for argument) is from 01 to 30
     - (Equal mark) is must
     - Max length for (displayed meaning of argument) is 15 characters, enclosed it double quotes.

   - In the case of V: Macro definition
     - Writing style is | V05 = "NOT-CONSECUTIVE"(macro definition name) :(colon) '1'(value) |
     - Max length for (macro definition name) is 15 characters.
     - Describe macro definitions as long as necessary after | =(equal mark) | with each macro definition separated by a | ,(comma) |.
     - The number of definable macros for an argument is up to 35.
     - The details are explained later in the section called "Macro definition".

   - In the case of W: List of string
     - Writing style is | W04 = "NO_WELD"(string) |
     - Max length is 34 characters for (string).
     - Describe strings as long as necessary after | =(equal mark) | with each string separated by a | ,(comma) |.
     - The number of string for an argument is up to 35.
     - Define the meaning of the string itself.

```
(Example)
 W04 = "NO_WELD", "WELD001", "WELD002", "WELDSTOP"
```

3. [ENDPROGRAM]
   - End.

4. For each additional program repeat step from 1 to 3.
   - The number of describable program is up to 1000.

5. Place a comment within braces if necessary { } .

Quote marks are used as follows.

```
< quote mark >
" "  : String
' '  : numerical value
```

The following codes are prohibited to use for String enclosed in quotes ″ ″:
- comma (,)
- semicolon (;)
- colon (:)
- single quotes (')
- double quotes (″)
- Line feed code
- Tab code

Macro Definition

Macros are used to define certain strings as a specific value.

For example, it is described in the text file as follows.

```
V03 = ″MAX_SPD″:'100', ″NORMAL_SPD″:'50', ″SLOW_SPD″:'10'
```

In this instance, '100' is passed as an argument if "MAX_SPD" is displayed in the field of the third argument on the EDIT screen.

For example,

```
-----------------------------------------------------------------------
[PROGRAM]
NAME = ″HANDLING″
ARGUMENT : '4'
[ARGUMENT]
N01 = ″LINE″
V02 = ″SLOW″:'1', ″FAST″:'2'
V03 = ″SMALL″:'1', ″NORMAL″:'2', ″BIG″:'3'
V04 = ″MAX_LOAD″:'10000', ″NO_LOAD″:'0'
[ENDPROGRAM]
-----------------------------------------------------------------------
```

If this text file is used, the following will be displayed on the Edit screen.

```
-----------------------------------------------------------------------
 (Example)
   [ Display ]   CALL HANDLING(LINE=3,SLOW,BIG,NO_LOAD)

   [ Process ]   CALL HANDLING(3,1,3,0)

-----------------------------------------------------------------------
```

The following integer is used for the macro:
-16777216 to 16777216 (-2^24 to 2^24)

A decimal number is unusable for the macro definition.

The same name or value cannot be used for an argument.

The number of definable macros for an argument is up to 35.

Change of Default Value

The default value for N(Constant number) is 0, for S(String) is '…'(null string).
You can change this default value for each argument.

In the case of N(Constant number), :(colon) where:

```
----------------------------------------------------------------------
[PROGRAM]      {OFFSSET}
NAME = "POS_OFFSET"
ARGUMENT = '3'
[ARGUMENT]
N01 = "X-OFFSET":'10'
N02 = "Y-OFFSET":'20'
N03 = "Z-OFFSET":'30'
[ENDPROGRAM]
----------------------------------------------------------------------
```

The specified default value is used when arguments are taught automatically, if you teach the call instruction while loading this text file.

```
1: CALL POS_OFFSET(X-OFFSET=10,Y-OFFSET=20,Z-OFFSET=30)
```

By default, integer numbers within the following range can be used:
-16777216 to 16777216 (-2^24 to 2^24)

Six significant figures are used in a decimal number.
For example, 123.456
If the default value is described over six significant figures, it is truncated.

Additionally, you can use the register or argument register as a default too. In that case, you describe :(colon) and (R) or (AR) as follows:

```
N01 = "X-OFFSET":'10'
N02 = "Y-OFFSET":(R)
N03 = "Z-OFFSET":(AR)
```

If you teach a call instruction in this state, a register or argument register is used as default as follows:

```
1: CALL POS_OFFSET(X-OFFSET=10,Y-OFFSET=R[...],
 : Z-OFFSET=AR[...])
```

In the case of S(String), you describe :(colon) as follows:

```
S01 = "Area Name":"Booth1"
```

If you teach the call instruction as well as the case of N(Constant number), the specified string is used as the default when arguments are taught automatically.

Additionally, you can also use the string register or argument register as the default. In that case, :(colon) and (SR) or (AR) are described as follows:

```
S01 = "Area Name":(SR)   or S01 = "Area Name":(AR)
```

File Name

File name is fixed as ARGDISPKN01.DT.
Where the configuration is "ARGDISP(fixed) + KN (language) + 01(serial number) + .DT(extension)"

ARGDISP(fixed) : Common part
KN (language) : language-specific where:
- EG : English
- KN : Japanese [Kanji]
- GR : German
- FR : French
- SP : Spanish
- CH : Chinese
- TW : Taiwanese
- CS : Czech
- RU : Russian
- OT : Other

01(serial number) : Usable from 01 to 99
It is possible to load different serial number files all together.
.DT(fixed extension) : Common part

> **NOTE**
> Japanese is not JP [Japanese] but KN [Kanji]. This is because there are two display styles for Japanese. JP is used for old software versions that use katakana. KN is used for current software versions where Kanji (2 byte character) is used.

How to load a Text File

Use the following procedure to load the text file.

1. Create the text file on the MC: (Memory Card).
2. Move the cursor to the file in the File screen, and select F3, [LOAD].
3. Cycle power.

Save and Load Text File as Backup

The text file is backed up and restored as follows:

(Backup)
- At the File Screen, select F4, [BACKUP] and “All of above” or “Application” in Cold start.

(Restore)
- At the File Screen, select F4, [RESTORE] and “All of above” or “Application” in Controlled start.

Alarm for Failure to Read Text File

If content of the loaded text file is not formatted correctly, it will not be read. An alarm such as “FILE-095 (ARGDISPKN01.DT) is not loaded.” is posted and recorded in the alarm history.

View the cause of the failure in related alarm messages displayed on the alarm history screen.

The following alarm messages can be displayed:

- FILE-095 (ARGDISPKN01.DT) is not loaded : File name to fail to read
- FILE-096 on line n, xxx : The line (n) and characters or value (xxx) to fail to read
- FILE-097 Out of memory : Out of Temporary memory to be required
- FILE-098 Over 1000 programs : Over the maximum programs
- FILE-099 Wrong end of file : End of file not after [ENDPROGRAM]
- FILE-100 Wrong tag/item/character code : Miss of tag, item or character code
- FILE-101 Over 35 definitions : Over the maximum definitions of macro or list of string
- FILE-102 Over limit value : Over the maximum value for macro definition or default value for Constant number

4.7.7.3 Editing

Use the following procedure to teach the call instruction arguments.

Teaching of Arguments

1. Input a CALL statement and select the program that is described in the text file.

```
1: CALL ...
[END]
```

2. Next, the arguments are taught automatically. (The default value is 0 or '…' (null string). For a macro definition or list of strings, the first item is set.)
   The argument descriptions are displayed in each field as follows.

```
1: CALL TRACKING(Area Name='...',VR num=0,
 : Timeout Time=0,Reg num timeout=0,NOT-CONSECUTIVE,
 : Model ID=0)
[END]
```

3. Change of value

The appearance is changed, but the input operation is not changed.

```
1: CALL TRACKING(Area Name='CStn_Out_R1', VR num=0,
 : Timeout Time=0,Reg num timeout=0,NOT-CONSECUTIVE,
 : Model ID=0)
[END]
```

Input value.

```
1: CALL TRACKING(Area Name='CStn_Out_R1', 2 ,
 : Timeout Time=0,Reg num timeout=0,NOT-CONSECUTIVE,
 : Model ID=0)
[END]
```

Push Enter.

```
1: CALL TRACKING(Area Name='CStn_Out_R1', VR num=2,
 : Timeout Time=0,Reg num timeout=0,NOT-CONSECUTIVE,
 : Model ID=0)
[END]
```

Submenu for Editing

The sub menu is displayed when F4, [CHOICE] is pressed while the cursor is on an argument. The submenu items displayed next are different.

1. Argument is N (Constant number)

   Constant, Register and Argument register are displayed as items on the submenu.

```
Parameter select 1
1 Constant
2 R[ ]
3 AR[ ]
4
5
6
7
8 --next page --
```

2. Argument is V (Macro definition)

Strings that are defined as macros are displayed as items on the submenu.

```
Parameter select 1
1 MAX_SPD
2 NORMAL_SPD
3 SLOW_SPD
4
5
6
7
8 --next page --
```

3. Argument is S (String)

String, String register and Argument registers are displayed as items on the submenu.

```
Parameter select 1
1 String
2 SR[ ]
3 AR[ ]
4
5
6
7
8 --next page --
```

4. Argument is W (list of String)

The list of strings is displayed as items on the submenu.

```
Parameter select 1
1 No Weld
2 Weld001
3 Weld002
4 Weld End
5
6
7
8 --next page --
```

NOTE

Up to 15 characters can be displayed as items on the submenu. And over 15 characters can be used for the list of string. If over 15 characters are used, the first 13 characters and '..' (two colons) are displayed as items on the submenu.
For example) ”PAINT_APPLICATION” -> “PAINT_APPLICA..”

Teaching Arguments Automatically

When you teach the CALL instruction and select the program that is written in the Text file, arguments are automatically set with the default value (See "Teaching of Arguments").
If you change the program name of the CALL instruction, arguments are taught. The default value is set for each argument also as follows.

```
1: CALL TRACKING(Area Name='CStn_Out_R1',VR num=2,
 : Timeout Time=(-1),Reg num timeout=1,NOT-CONSECUTIVE,
 : Model ID=1)
```

↓ Change the program name from TRACKING to POS_OFFSET

```
1: CALL POS_OFFSET(X-OFFSET=10,Y-OFFSET=20,Z-OFFSET=30)
```

If the called program name is changed to the another program name that is not described in text file, Wizard to input arguments is disabled and the previous argument value(s) will be restored.

4.7.7.4 Handling of Multiple Languages

The text file (DT file) is needed to be prepared for each language. The text file is read by the current language setting on start-up. If the current language is switched and the text files of multiple languages is loaded, controller power must be cycled for the new language file to be read.

The Text file for the current language must exist for Wizard to input arguments to operate properly.

4.7.7.5 Save and Upload of ASCII Files

If you save a TP program as an ASCII file (LS file) when Wizard to input arguments is enabled, the meaning of argument is enclosed by " "(double quotes). [For example : VR num=2 > "VR num"=2].
And in case of macro definition, not only macro definition name is enclosed by " ", but also the defined value is described as | =(value) |. [For example : NOT-CONSECUTIVE > "NOT-CONSECUTIVE"=1]

```
1: CALL TRACKING(Area Name='CStn_Out_R1',VR num=2,
 : Timeout Time=(-1),Reg num timeout=1,NOT-CONSECUTIVE,
 : Model ID=1)
```

↓ Save as ASCII file

```
1: CALL TRACKING("Area Name"='CStn_Out_R1',"VR num"=2,
 : "Timeout Time"=(-1),"Reg num timeout"=1,
 : "NOT-CONSECUTIVE"=1,"Model ID"=1)
```

You can upload this ASCII file if the ASCII upload option is loaded. The argument description enclosed in quotes (" ") is ignored when the ASCII file is uploaded.
Therefore, you can upload the ASCII file without considering whether the language is set correctly or the text file is loaded in advance.

> **NOTE**
> Wizard to input arguments and the process to ignore the argument description enclosed by quotes (" ") is supported only in software version 7DC3(V8.30) and later. An ASCII file that includes an argument description enclosed in quotes (“ “) will cause the process to fail in versions released before software version 7DC3(V8.30).

4.7.7.6 Mode Selection

You can change the mode of Wizard to input arguments using the system variable $ARGDISPMODE, as follows:

0 : Wizard to input arguments is disabled.
1 : Wizard to input arguments is enabled. (Default)
2 : Wizard to input arguments is enabled and the value of macro definition is displayed.

```
(Example)
   [ $ARGDISPMODE=0 ]   CALL HANDLING(3,1,3,0)

   [ $ARGDISPMODE=1 ]   CALL HANDLING(LINE=3,SLOW,BIG,NO_LOAD)

   [ $ARGDISPMODE=2 ]   CALL HANDLING(LINE=3,SLOW=1,BIG=3,NO_LOAD=0)
```

The change of the mode is reflected after power is cycled.

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  1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Palletizing
8 --next page--
```

- Time-specified wait instruction: Waits program execution for a specified period of time.
- Conditional wait instruction: Waits program execution until a specified condition is satisfied or a specified period of time has elapsed.

4.8.1 Time-specified Wait Instruction

WAIT (TIME)

The time-specified wait instruction waits program execution for a specified period of time (in seconds).

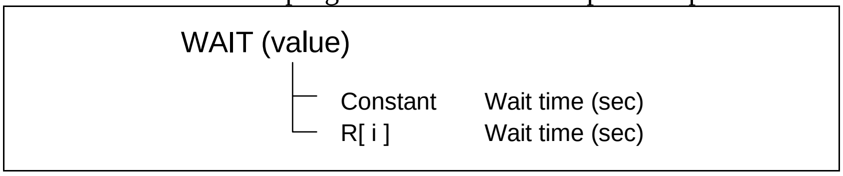


Fig. 4.8.1 Time-specified wait instruction

Example
```
1: WAIT
2: WAIT 10.5sec
3: WAIT R[1]
```

4.8.2 Conditional Wait Instructions

WAIT (condition) (processing)

A conditional wait instruction waits program execution until a specified condition is satisfied or a specified period of time has elapsed. Two methods of specifying time-out processing are available:

- If no processing is specified, program execution waits until a specified condition is satisfied.
- Timeout, LBL[i] is transferred to a specified label if the specified condition is not satisfied until the time specified in “14 WAIT timeout” on the system configuration screen.

Register conditional wait instruction

The register conditional wait instruction compares the value of a register with another value, and waits until the comparison condition is satisfied,

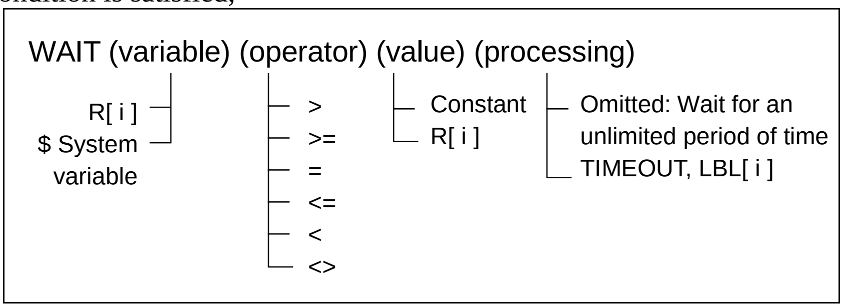


Fig. 4.8.2 (a) Register conditional wait instruction

Example
```
3: WAIT R[2] <> 1, TIMEOUT LBL[1]
4: WAIT R[ R[1] ]> = 200
```

I/O conditional wait instruction

The I/O conditional wait instruction compares the value of an input/output signal with another value, and waits until the comparison condition is satisfied.

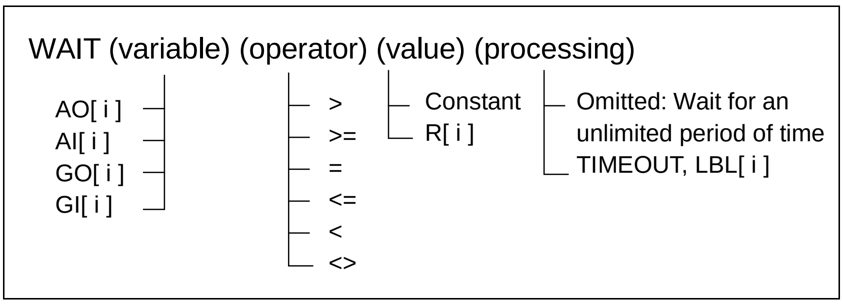


Fig. 4.8.2 (b) I/O conditional wait instruction 1

```
WAIT (variable) (operator) (value) (processing)

(variable):   DO[ i ], DI[ i ], RO[ i ], RI[ i ], SO[ i ], SI[ i ], UO[ i ], UI[ i ]
(operator):   =, <>
(value):      ON, OFF, DO[ i ], DI[ i ], RO[ i ], RI[ i ], On+ (Note), Off- (Note),
              SO[ i ], SI[ i ], UO[ i ], UI[ i ], R[ i ] : 0 Off, 1 On
(processing): Omitted: Wait for an unlimited period of time
              TIMEOUT, LBL[ i ]
```

Fig. 4.8.2 (c) I/O conditional wait instruction 2

Example
```
5: WAIT DI[2] <> OFF, TIMEOUT LBL[1]
6: WAIT RI[ R[1] ] = R[1]
```

NOTE

Off-:
The falling edge of a signal is regarded as being a detection condition. The condition is not satisfied while the signal remains off. The detection condition is satisfied when the signal changes from the on state to the off state.
On+:
The rising edge of a signal is regarded as being a detection condition. The condition is not satisfied while the signal remains on. The detection condition is satisfied when the signal changes from the off state to the on state.

Error condition wait instruction

The error condition wait instruction waits for the occurrence of an alarm having a specified error number.

```
WAIT ERR_NUM = (value) (processing)

(value):      Constant (Note)
(processing): Omitted: Wait for an unlimited period of time
              TIMEOUT, LBL[ i ]
```

Fig. 4.8.2 (d) Error condition wait instruction

NOTE

An error number is specified with an alarm ID followed by an alarm number.
Error number = aabbb
where aa = alarm ID
bbb = alarm number
For an explanation of alarm IDs and numbers, refer to “FANUC Robot series OPERATOR’S MANUAL (Alarm Code List)” (B-83284EN-1).

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>

4.8.3 The Output When Wait on Input

The Output When Wait on Input is the function that outputs DO when the timeout time is passed during WAIT instruction of program is waiting for specified DI. By this function, DI and GI can be observed.

To use this function, set the range of the observed input signal, the timeout time and the output signal. The conditions of the observed wait instruction are as follows.

- DI[ ] = ON
- DI[ ] = OFF
- DI[ ] = On+
- DI[ ] = Off-
- GI[ ] = Constant
- GI[ ] <> Constant
- GI[ ] < Constant
- GI[ ] <= Constant
- GI[ ] > Constant
- GI[ ] >= Constant

Table 4.8.3 Output when wait on input function setup item

Setup item	Description
Wait for DI range	Set the Output When Wait on DI function by setting the following three items. When the timeout time specified in “Wait for DI time” is passed during WAIT instruction of program is waiting for DI signal specified in “Wait for DI range”, the output signal specified in “Wait for DI output” is turned on. In this setup item, specify the DI number to observe.
Wait for DI time	Specify the timeout period for the Output When Wait on DI.
Wait for DI output	Specify the output signal number for the Output When Wait on DI.
Wait for GI range	Set the Output When Wait on GI function by setting the following three items. When the timeout time specified in “Wait for GI time” is passed during WAIT instruction of program is waiting for GI signal specified in “Wait for GI range”, the output signal specified in “Wait for GI output” is turned on. In this setup item, specify the GI number to observe.

Setup item	Description
Wait for GI time	Specify the timeout period for the Output When Wait on GI.
Wait for GI output	Specify the output signal number for the Output When Wait on GI.

Procedure 4-4 Setting of Output When Wait On Input Function

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select “6 SYSTEM” in the next page.
3 Press F1, [TYPE]. The screen change menu will be displayed.
4 Select "Config". The system configuration screen will be displayed.

```
System/Config
                                              1/47
      1 Use HOT START:                       TRUE
      2 I/O power fail recovery: RECOVER ALL
      3 COLD START Autoexec program:
       [************************************]
      4 HOT START Autoexec program:
       [************************************]
      5 HOT START done signal:      DO[    0]
      6 Restore selected program:      TRUE
      7 Enable UI signals:             TRUE
      8 START for CONTINUE only:       FALSE
      9 CSTOPI for ABORT:              FALSE

   [ TYPE ]                   TRUE     FALSE
```

5 Move the cursor to <**DETAIL**> of the item “Output when WAIT on input”, and press the [ENTER] key.

```
System/Config
                                             40/47
     37 Sim. Input Wait Delay:        0.00sec
     38 Set if SIm. Skip Enabled:   DO[    0]
     39 Set when prompt displayed:  DO[    0]
     40 Output when WAIT on Input:<**DETAIL**>
     41 Signal if OVERRIDE = 100    DO[    0]
     42 Hand broken :              <*GROUPS*>
     43 Remote/Local setup:    Local
     44    External I/O(ON:Remote): DI[    0]
     45 UOP auto assignment:     ALL
     46 Multi Program Selection:       FALSE
     47 WAIT at Taught Position:       FALSE

   [ TYPE ]
```

6 The Output When Wait on Input function setup screen is displayed.

```
System/Config
   Output when WAITing on Input        1/6

   1  WAIT for DI range:  DI[     0 -      0]
   2  WAIT for DI time:             0.00 sec
   3  WAIT for DI output:         DO[      0]
   4  WAIT for GI range:  GI[     0 -      0]
   5  WAIT for GI time:             0.00 sec
   6  WAIT for GI output:         DO[      0]

   Exit menu to apply changes

   [ TYPE ]
```

7 After the setup is finished, press [PREV] key to display to the system configuration screen. When the system configuration screen is displayed, the setting data are available.

4.9 SKIP CONDITION INSTRUCTION

The skip condition instruction specifies, in advance, a skip condition (condition for executing a skip instruction) used with a skip instruction. Before a skip instruction can be executed, a skip condition instruction must be executed. A skip condition once specified is valid until the execution of the program is completed, or the next skip condition instruction is executed.

```
  Instruction  2
1 Miscellaneous
2 Skip
3 Payload
4 Offset/Frames
5 Multiple control
6 Program control
7 MACRO
8 --next page--
```

A skip instruction causes a jump to a branch destination label if the skip condition is not satisfied. If the skip condition is satisfied, a skip instruction causes the robot to suspend the current motion toward a target point, instead executing the program instruction on the next line.
If the skip condition is currently not satisfied, a skip instruction causes a jump to a destination label upon the completion of the current motion.

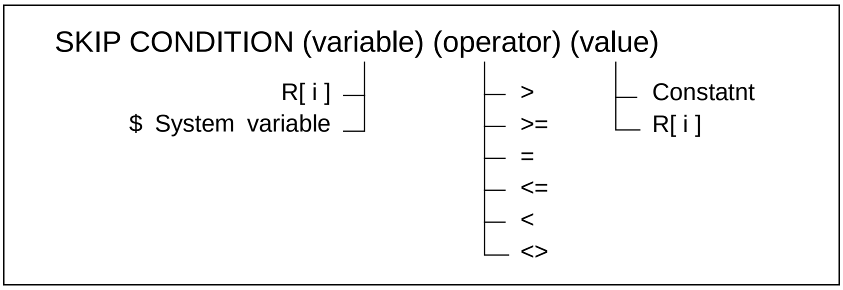


Fig. 4.9 (a) Skip condition instruction (register condition)

```
SKIP CONDITION (variable) (operator) (value)
          AO[ i ]          >        Constant
          AI[ i ]          >=       R[ i ]
          GO[ i ]          =
          GI[ i ]          <=
                           <
                           <>
```

Fig. 4.9 (b) Skip condition instruction (I/O condition 1)

```
SKIP CONDITION (variable) (operator) (value)
          DO[ i ]          =        ON
          DI[ i ]          <>       OFF
          RO[ i ]                   DO[ i ]
          RI[ i ]                   DI[ i ]
          SO[ i ]                   RO[ i ]
          SI[ i ]                   RI[ i ]
          UO[ i ]                   On+ (Note)
          UI[ i ]                   Off- (Note)
                                    SO[ i ]
                                    SI[ i ]
                                    UO[ i ]
                                    UI[ i ]
                                    R[ i ] : 0 Off, 1 On
```

Fig. 4.9 (c) Skip condition instruction (I/O condition 2)

Example
```
1: SKIP CONDITION DI[ R[1] ] <> ON
2: J  P[1] 100% FINE
3: L  P[2] 1000mm/sec FINE Skip, LBL[1]
4: J  P[3] 50% FINE
5:    LBL[1]
6: J  P[4] 50% FINE
```

NOTE

Off-:
The falling edge of a signal is regarded as being a detection condition. The condition is not satisfied while the signal remains off. The detection condition is satisfied when the signal changes from the on state to the off state.

On+:
The rising edge of a signal is regarded to be a detection condition. The condition is not satisfied while the signal remains on. The detection condition is satisfied when the signal changes from the off state to the on state.

```
SKIP CONDITION ERR_NUM = (value)
                            Constant (Note)
```

Fig. 4.9 (d) Skip condition instruction (error condition)

NOTE

An error number is specified with an alarm ID followed by an alarm number.
Error number = aabbb
where aa = alarm ID
bbb = alarm number
For an explanation of alarm IDs and numbers, refer to “FANUC Robot series OPERATOR’S MANUAL (Alarm Code List)” (B-83284EN-1).
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 PAYLOAD INSTRUCTION

Payload instruction is the instruction to switch the payload data (payload schedule number). If the payload which is mounted on the robot is changed during the program execution by loading/unloading the workpiece or attaching/detaching the tool, and so on, please switch the payload data correctly by this instruction.
In order to execute this instruction, it is required to set up the payload data in advance. Refer to the Section 3.17 "PAYLOAD SETTING" about the method to set up the payload data.

```
   Instruction   2
1 Miscellaneous
2 Skip
3 Payload
4 Offset/Frames
5 Multiple control
6 Program control
7 MACRO
8 --next page--
```

PAYLOAD [i]

This instruction activates payload schedule No. i.
In case that the payload schedule is switched by this instruction in a TP program, even after the TP program execution is finished, the selected schedule keeps active. In other words, once schedule number is changed, the setting will be used for program execution and jog operation afterwards.

```
PAYLOAD [ i ]
           ├─ R[ i ]
           └─ Constant
              Payload schedule number (1~10)
```

Fig. 4.10 Payload instruction

Example

```
1:  PAYLOAD[1]:
```

This program activates payload schedule No. 1.

Multi-group system

In multi-group system, PAYLOAD[i] instruction activates the payload schedule No. i for all motion groups which belong to the program. If you want to mask the target group, use PAYLOAD[GPk : i] instruction as follows;

1 Add PAYLOAD[i] instruction.

```
PROGRAM
                                          1/2
    1:   PAYLOAD[...]
   [End]


  Enter value
   GP_MASK  DIRECT  INDIRECT
```

2 Put the cursor on [...] and press F1, "GP_MASK". Select “2 [GP:]” in the popup menu, then the format changes to PAYLOAD[GPk : i].

```
PROGRAM
                                          1/2
    1:   PAYLOAD[GP1,2,3:...]
   [End]



    GP1     GP2     GP3
```

3 Select the target group numbers and enter payload schedule number.

Example

```
1:  PAYLOAD[GP2,3:1];
```

This program activates payload schedule No. 1 for Group 2 and Group 3.

4.11 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 Subsection 4.3.5 Additional Motion Instructions.)

```
   Instruction  2
1 Miscellaneous
2 Skip
3 Payload
4 Offset/Frames
5 Multiple control
6 Program control
7 MACRO
8 --next page--
```

- The position register specifies the shifting direction and the shift amount.
- When the positional information is expressed in the joint frame, the shift amount of each axis is applied.
- When the positional information is expressed in the Cartesian coordinate system, the number of user frame by which the offset condition is decided should be specified. When it is not specified, the user frame being selected now is used.

> ⚠ **CAUTION**
> If teaching is made by joint coordinates, changing the user coordinate system does not affect the position variables and position registers. However, note that both position variables and registers are affected by the user coordinate systems when the robot is taught in the Cartesian format.

The OFFSET instruction shifts positional information programmed at the destination position by the offset amount specified by position register, and moves the robot to the shifted position.
The shifting condition is specified by the OFFSET CONDITION instruction.

OFFSET CONDITION PR[i] (UFRAME[j])

Position register number (1 to 100)

User frame number (1 to 9)

Fig. 4.11 Offset conditional instruction

Example
```
1:  OFFSET CONDITION PR[ R[1] ]
2: J  P[1] 100% FINE
3: L  P[2] 500mm/sec FINE Offset
```

4.12 TOOL OFFSET CONDITION INSTRUCTIONS

A tool offset condition instruction specifies the offset condition used in a tool offset instruction. Execute a tool offset condition instruction before executing the corresponding tool offset instruction. Once the tool offset conditions have been specified, they remain effective until the program terminates or the next tool offset condition instruction is executed. (For the tool offset instruction, see Subsection 4.3.5 ”Additional motion instructions”.)

```
 Instruction  3
1 FOR/ENDFOR
2 Tool_Offset
3 LOCK PREG
4 MONITOR/MON. END
5 String
6
7
8 --next page--
```

- The position register specifies the direction in which the target position shifts, as well as the amount of shift.
- The tool coordinate system is used for specifying offset conditions.
- When the number of a tool coordinate system is omitted, the currently selected tool coordinate system is used.
- When the position data is expressed in joint coordinate system, 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.

```
TOOL_OFFSET CONDITION PR[ i ] (UTOOL[ j ])
    Position register number (1 to 100) ─┘        │
               Tool frame number (1 to 10) ───────┘
```

Fig. 4.12 Tool offset condition instruction

Example
```
1: TOOL_OFFSET PR[1]
2: J P[1] 100% FINE
3: L P[2] 500mm/sec FINE Tool_Offset
```

4.13 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  2
1 Miscellaneous
2 Skip
3 Payload
4 Offset/Frames
5 Multiple control
6 Program control
7 MACRO
8 --next page--
```

- 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.

```
UTOOL[ i ] = (value)
Tool frame number (1 to10) ─┘          └─PR[ i ]
```

Fig. 4.13 (a) Tool frame setup instruction

```
UFRAME[ i ] = (value)
User frame number (1 to 9) ─┘          └─PR[ i ]
```

Fig. 4.13 (b) User frame setup instruction

Example
```
1: TOOL[1] = PR[1]
2: UFRAME[3] = PR[2]
```

Frame select instruction

The tool frame select instruction changes the current tool frame number.
The user frame select instruction changes the current user frame number.

```
UTOOL_NUM =(value)
            ├─ R[ i ]
            └─ Constant      Tool frame number (1 to 10)
```

Fig. 4.13 (c) Tool frame select instruction

```
UFRAME_NUM = (value)
             ├─ R[ i ]
             └─ Constant      User frame number (1 to 9)
```

Fig. 4.13 (d) User frame select instruction

Example
```
1:  UFRAME_NUM = 1
2: J P[1] 100% FINE
3: L P[2] 500mm/sec FINE
4:  UFRAME_NUM = 2
5: L P[3] 500mm/sec FINE
6: L P[4] 500mm/sec FINE
```

4.14 PROGRAM CONTROL INSTRUCTIONS

The program control instructions control program execution.

```
   Instruction   2
1 Miscellaneous
2 Skip
3 Payload
4 Offset/Frames
5 Multiple control
6 Program control
7 MACRO
8 --next page--
```

- Pause instruction
- Abort instruction

4.14.1 Pause Instruction

PAUSE

The pause 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.

```
PAUSE
```

Fig. 4.14.1 Pause instruction

4.14.2 Abort Instruction

ABORT

The abort instruction aborts program execution in the following way, causing the robot in motion to decelerate and stop:

- If an operation instruction is being executed, the program stops before the operation is completed.
- The cursor stops on the current line.
- When the abort instruction is executed, the execution of the program cannot be continued. Information held by a program call instruction about the main program is lost.

```
ABORT
```

Fig. 4.14.2 Abort instruction

4.15 OTHER INSTRUCTIONS

The following miscellaneous instructions are available:

```
   Instruction  2
1 Miscellaneous
2 Skip
3 Payload
4 Offset/Frames
5 Multiple control
6 Program control
7 MACRO
8 --next page--
```

- RSR instruction
- User alarm instruction
- Timer instruction
- Override instruction
- Comment instruction
- Message instruction
- Parameter instruction
- Maximum speed instruction

4.15.1 RSR Instruction

RSR [i] = (value)

The RSR instruction alternately enables and disables the RSR function having a specified RSR number.

```
              RSR[ i ] = (value)
RSR signal number ┘      ├─ Enable :  Enable RSR function
     (1 to 4)            └─ Disable : Disable RSR function
```

Fig. 4.15.1 RSR instruction

Example 1: **RSR[2:Workproc.2.]=ENABLE**

4.15.2 User Alarm Instruction

UALM[i]

The user alarm instruction displays the alarm message corresponding to an already set user alarm number on the alarm display line. The user alarm instruction pauses the program which is on progress. A user alarm is specified on the user alarm setting screen (See Section 3.12) and this setting is registered in the system variable $UALRM_MSG . The total number of user alarms can be changed at a controlled start (See Section B.1, START MODE).

```
UALM[ i ]
       └─ Alarm number
```

Fig. 4.15.2 User alarm instruction

Example **1: UALM[1] ($UALRM_MSG[1] = WORK NOT FOUND**

$UALRM_MSG is saved to SYSVARS.SV.

4.15.3 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].

```
     TIMER[ i ] = (processing)
Timer number ┘      ├─ START  : Start the timer.
                    ├─ STOP   : Stop the timer.
                    ├─ RESET  : Reset the timer.
                    │         : Clear the timer overflow flag.
                    └─ (…)    : Load the value into the timer,
                                then strat the timer.
                              : Clear the timer overflow flag.
```

Fig. 4.15.3 Timer instruction

Example
```
1: TIMER [1]=START
   TIMER [1]=STOP
   TIMER [1]=RESET
   TIMER [1]=(R[1]+1)
```

The mixed logic instruction can be used as the instruction to load the value into the timer. (Refer to the section 4.19 MIXED LOGIC INSTRUCTION.)

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[1]=TIMER_OVER FLOW[1]
0: Not over flow
1: Over flow

When the instruction to start timer which has already been started, the instruction to start timer is ignored. In this case, the warning “INTP-685 TIMER[ ] has already been started” is issued.

In case that the timer is set to the local timer, the timer can be handled by only one task. If the instruction to stop the timer is executed in another task than the task which has started the timer, the timer is not stopped. If you want to handle a timer in the multiple tasks, set the timer to the global timer in the program timer screen. (Refer to the section 7.9 PROGRAM TIMER.)

4.15.4 Override Instruction

OVERRIDE = (value)%

The override instruction changes a feed rate override.

OVERRIDE = (value) %

- R[i]
- Constatnt
- AR[i]

(value) : Feedrate override(1 to 100)

Fig. 4.15.4 Override instruction

Example `1: OVERRIDE = 100%`

4.15.5 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.

! (Remark)

- A comment can consist of up to 32 characters including alphanumeric characters, asterisks(*), underlines(_), at marks(@), etc.

Fig. 4.15.5 Comment instruction

Example `1: !APPROACH POSITION`

4.15.6 Multi-language Comment Instruction

--(Remark)

The multi-language comment instruction adds a comment in a program alike to the comment instruction. A comment has no effect on program execution. To add a comment, press the [ENTER] key.
It is different from 4.15.5 Comment Instruction. It is possible to input comment to each language independently. (Refer to 3.16 SETTING THE GENERAL ITEMS about change current language.)

```
-- (Remark)
      └─ A comment can consist of the characters including
         alphanumeric characters, asterisks(*), underlines(_),
         at marks(@), etc.
```

Fig. 4.15.6 Multi-language comment instruction

Example

```
1: --APPROACH POSITION          (Current language : ENGLISH)
                        ↓  switch to language
1: --                           (Current language : JAPANESE)
```

In this example, the comment in English is not reflected to Japanese comment, because the comment is independent to each language. It is necessary to input the comment to each language.

4.15.7 Message Instruction

MESSAGE[message statement]

The message instruction displays a specified message on the user screen. (For the user screen, see the section 7.2.) A message can consist of up to 24 characters including alphanumeric characters, asterisks (*), underlines (_) , and at marks (@). To add a comment, press the [ENTER] key.

```
MESSAGE   [message statement]
              └─ A message statement can consist of up to 24
                 characters including alphanumeric characters,
                 asterisks(*), underlines(_), at marks(@), etc.
```

Fig. 4.15.7 Message instruction

Example `1: MESSAGE[ DI[1] NOT INPUT ]`

4.15.8 Parameter Instruction

$(SYSTEM VARIABLE NAME) = (value)

The parameter instruction changes the value of a system variable. This instruction can be used only for a system variable containing a numeric value (constant). You can enter the parameter name after pressing the [ENTER] key. It is possible to enter the parameter name up to 30 characters or less without the first character,"$".
There are two types of system variables, variable type and position type. A system variable of variable type can be assigned to a register. A system variable of position type can be assigned to a position register.
System variables of position data type are divided into three data types, Cartesian (XYZWPR type), joint type (J1-J6 type), and matrix type (AONL type).
When a system variable of position data type is assigned to a position register, the data type of the position register is converted to the data type of the system variable.

If a system variable of position type is assigned to a register, or if a system variable of variable type is assigned to a position register, the following alarm is generated during execution.

INTP-240 Incompatible data type

```
$(SYSTEM VARIABLE name) = (value)
System variable name             System variable value
                                 (numeric value)
                                 R[X]
                                 PR[X]
```

Fig. 4.15.8 (a) Parameter instruction (writing)

Example **`1: $SHELL_CONFIG.$JOB_BASE = 100`**

```
(value) = $(SYSTEM VARIABLE name)
R[X]               System variable name
PR[X]
```

Fig. 4.15.8 (b) Parameter instruction (reading)

Example **`1: R[1] = $SHELL_CONFIG.$JOB_BASE`**

> ⚠ **WARNING**
> The operation of the robot and controller is controlled with system variables. Only a person who is aware of how changes to the system variables will affect the system should set system variables. If a person without detailed knowledge attempts to set the system variables, the robot and controller would malfunction, causing injury to personnel or damage to equipment.

4.15.9 Maximum Speed Instructions

A maximum speed instruction specifies the maximum operating speed of a program. There are two maximum speed instructions, the instruction for specifying the joint operation speed and that for specifying the path control operating speed. If a speed exceeding the speed specified with a maximum speed instruction is specified, the speed specified with the maximum speed instruction is assumed.

JOINT_MAX_SPEED[i]=(value)

```
JOINT_MAX_SPEED [ i ] = (value)
Axis number(1 to 9)        Constatnt (deg/sec)
                           R[ i ]    (deg/sec)
```

Fig. 4.15.9 (a) Joint max speed instruction

Example **`JOINT_MAX_SPEED[3] = R[3]`**

LINEAR_MAX_SPEED= (value)

```
LINEAR_MAX_SPEED = (value)
                      ├─ Constatnt  (deg/sec)
                      └─ R[ i ]     (deg/sec)
```

Fig. 4.15.9 (b) Path control max speed instruction

Example `LINEAR_MAX_SPEED = 100`

For following case, specified maximum speed is changed to default value.
- Run aborted program.
- Run aborted program by Backward execution
- Abort program.
- Call any programs by CALL instruction.

4.16 MULTIAXIS CONTROL INSTRUCTIONS

Multiaxis control instructions control the execution of a multitask program.

```
   Instruction    2
1 Miscellaneous
2 Skip
3 Payload
4 Offset/Frames
5 Multiple control
6 Program control
7 MACRO
8 --next page--
```

- Program execution instruction

4.16.1 Program Execution Instruction

During the execution of a program, the program execution instruction starts the execution of another program.

- The difference from the program call instruction is that, with the program call instruction, those lines following the call instruction are executed after the called program has been executed, whereas with the program execution instruction, the program that starts the execution of another program continues concurrently.
- To synchronize programs that are being executed simultaneously, use the register instruction and the register condition wait instruction.
- If an attempt is made to execute a program for which the same motion group is specified, an alarm is generated. If this occurs, specify a different motion group.

```
RUN   (program name)
          └─ Name of a program to be run
```

Fig. 4.16.1 Program execution instruction

Example PROG1	PROG2
1: R[1]=0	1: J P[3] 100% FINE
2: RUN PROG2	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 R1[1]=1	5: R[1]=1
MOTION GROUP[1,*,*,*,*,*,*,*]	MOTION GROUP[*,1,*,*,*,*,*,*]

4.17 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 circular motion and circle arc motion)
- Specification of the feed rate 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  3
1 FOR/ENDFOR
2 Tool_Offset
3 LOCK PREG
4 MONITOR/MON. END
5 Independent GP
6 Simultaneous GP
7 String
8 --next page--
```

- 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, feed rate, 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.

4.17.1 Asynchronous Operation Group Instruction

The asynchronous operation group instruction controls operation groups asynchronously, with the operation formats, feed rates, and positioning formats specified separately for the individual operation groups.

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 operartion group

Fig. 4.17.1 Asynchronous operation group instruction

4.17.2 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 feed rate 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.

```
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 operartion group

Fig. 4.17.2 Synchronous operation group instruction

4.18 FOR/ENDFOR INSTRUCTION

FOR/ENDFOR statement is a function that repeats a loop within FOR and ENDFOR statements specified times.

```
    Instruction  3
1 FOR/ENDFOR
2 Tool_Offset
3 LOCK PREG
4 MONITOR/MON. END
5 String
6 DIAGNOSE
7
8 --next page--
```

FOR/ENDFOR statement has two instructions, FOR statement and ENDFOR statement.

- FOR statement start of FOR/ENDFOR loop
- ENDFOR statement end of FOR/ENDFOR loop

Instructions within FOR and ENDFOR statements are repeated.
The number of times to repeat is determined by the specified values in FOR statement.

4.18.1 FOR Statement

The form of FOR statement is as follows.

```
FOR (Loop counter) = (Initial value) TO (Target value)
```

- (Loop counter): R[i]
- (Initial value): Constant, R[i], AR[i]
- (Target value): Constatnt, R[i], AR[i]

Fig. 4.18.1 (a) FOR statement (When TO is selected.)

```
FOR (Loop counter) = (Initial value) DOWNTO (Target value)
          └─ R[ i ]          ├─ Constant        ├─ Constatnt
                             ├─ R[ i ]          ├─ R[ i ]
                             └─ AR[ i ]         └─ AR[ i ]
```

Fig. 4.18.1 (b) FOR statement (When DOWNTO is selected.)

- Register is used for Loop counter.
- Constant, register, argument register is used for Initial value. For constant, integer whose range is -32767 to 32766 can be specified.
- Constant, register, argument register is used for Target value. For constant, integer whose range is -32767 to 32766 can be specified.

When For statement is executed, Initial value is substituted into Loop counter. The following condition should be satisfied to execute FOR/ENDFOR loop.

- When TO is specified, Initial value is equal to or smaller than Target value.
- When DOWNTO is specified, Initial value is equal to or larger than Target value.

When this condition is not satisfied, the cursor moves to the next line of combined ENDFOR statement and FOR/ENDFOR loop is not executed.
FOR statement is executed only once in the FOR/ENDFOR loop.

4.18.2 ENDFOR Statement

The form of ENDFOR statement is as follows.

```
ENDFOR
```

Fig. 4.18.2 ENDFOR statement

FOR/ENDFOR loop is repeated as long as the following condition is satisfied.

- When TO is specified, the value of Loop counter is smaller than Target value.
- When DOWNTO is specified, the value of Loop counter is larger than Target value.

When this condition is satisfied and TO is specified, the value of Loop counter is incremented. When this condition is satisfied and DOWNTO is specified, the value of Loop counter is decremented. And after that, the cursor moves to the next line of combined FOR statement.
When this condition is not satisfied, the cursor moves to the next line and FOR/ENDFOR loop is not repeated.

> **NOTE**
> An internal delay is used when FOR/ENDFOR loop is executed repeatedly. Therefore, WAIT statement is not necessary in FOR/ENDFOR loop to repeat the loop.

4.18.3 FOR/ENDFOR Statement Combination

FOR and ENDFOR statements are automatically combined after teaching. The closest FOR statement and ENDFOR statement is orderly combined.
By teaching FOR/ENDFOR statements additionally in FOR/ENDFOR loop, nested loops can be formed. Up to 10 nested loops can be formed. However, teaching more than 10 nested loops causes an alarm in execution.
The number of FOR statement and ENDFOR statement in a program should be the same. When the number is not the same, an alarm occurs in execution, Alarms related to FOR/ENDFOR function is described in section 4.18.6.
How to combine FOR and ENDFOR statements is described in the following example.

Teach FOR statement in line 1. In this case, the number of FOR statement and ENDFOR statement is not the same. Therefore, if this program is executed, the alarm “INTP-670 Need ENDFOR for FOR in line 1” occurs.

```
PROGRAM
                                          1/8
       1:   FOR R[1]=1 TO 5
       2: L  P[1] 100mm/sec CNT100
       3:
       4: L  P[2] 100mm/sec FINE
       5: L  P[3] 100mm/sec CNT100
       6:
       7: L  P[4] 100mm/sec CNT100
     [End]

   [ INST ]                        [EDCMD]  >
```

Teach ENDFOR statement in line 6. FOR/ENDFOR loop is formed by FOR statement in line 1 and ENDFOR statement in line 6.

```
PROGRAM
                                         6/8
       1:   FOR R[1]=1 TO 5
       2: L  P[1] 100mm/sec CNT100
       3:
       4: L  P[2] 100mm/sec FINE
       5: L  P[3] 100mm/sec CNT100
       6:   ENDFOR
       7: L  P[4] 100mm/sec CNT100
     [End]

   [ INST ]                        [EDCMD]  >
```

Teach FOR statement in line 3. In this case, the number of FOR statement and ENDFOR statement is not the same. Therefore, if this program is executed, the alarm “INTP-670 Need ENDFOR for FOR in line 1” occurs.

```
PROGRAM
                                        3/8
      1:   FOR R[1]=1 TO 5
      2: L  P[1] 100mm/sec CNT100
      3:   FOR R[2]=5 DOWNTO 1
      4: L  P[2] 100mm/sec FINE
      5: L  P[3] 100mm/sec CNT100
      6:   ENDFOR
      7: L  P[4] 100mm/sec CNT100
      8:
  [End]

 [ INST ]                    [EDCMD]  >
```

Teach ENDFOR statement in line 8. The closest FOR statement and ENDFOR statement is orderly combined. Therefore, the first FOR/ENDFOR loop is formed by FOR statement in line 3 and ENDFOR statement in line 6, and second FOR/ENDFOR loop is formed by FOR statement in line 1 and ENDFOR statement in line 8. In this example, 2 nested loops are formed.

```
PROGRAM
                                        8/8
      1:   FOR R[1]=1 TO 5
      2: L  P[1] 100mm/sec CNT100
      3:   FOR R[2]=5 DOWNTO 1
      4: L  P[2] 100mm/sec FINE
      5: L  P[3] 100mm/sec CNT100
      6:   ENDFOR
      7: L  P[4] 100mm/sec CNT100
      8:   ENDFOR
  [End]

 [ INST ]                    [EDCMD]  >
```

> **NOTE**
> Please be careful not to use the same register number as Loop counters in the same nested loop. It may cause abnormal behavior.

4.18.4 Backward Execution of FOR/ENDFOR Statement

Backward execution is prohibited on FOR/ENDFOR statement.
However, backward execution on instructions within FOR/ENDFOR loop is allowed.
In the following example, when backward execution is started from line 1 or 5, the alarm “INTP-238 BWD execution completed” occurs. On the other hand, backward execution can be done when started from line 2, 3, 4 or 6.

```
PROGRAM
                                          1/7
      1:   FOR R[1]=1 TO R[2]
      2: L  P[1] 100mm/sec CNT100
      3: L  P[2] 100mm/sec CNT100
      4: L  P[3] 100mm/sec FINE
      5:   ENDFOR
      6: L  P[4] 100mm/sec CNT100
     [End]

   [ INST ]                      [EDCMD]  >
```

4.18.5 Examples of FOR/ENDFOR Statement Execution

The example of FOR/ENDFOR statement is described with the following program.

```
PROGRAM
                                          1/7
      1:   FOR R[1]=1 TO R[2]
      2: L  P[1] 100mm/sec CNT100
      3: L  P[2] 100mm/sec CNT100
      4: L  P[3] 100mm/sec FINE
      5:   ENDFOR
      6: L  P[4] 100mm/sec CNT100
     [End]

   [ INST ]                      [EDCMD]  >
```

Case 1: R[2]=3

TO is specified and Initial value is smaller than Target value. Therefore, the condition of FOR statement is satisfied.

As the value of Loop counter changes from 1 to 3, the condition of ENDFOR statement is satisfied and FOR/ENDFOR loop repeated three times.

```
PROGRAM
                                          1/7
      1:   FOR R[1]=1 TO R[2]
      2: L  P[1] 100mm/sec CNT100
      3: L  P[2] 100mm/sec CNT100
      4: L  P[3] 100mm/sec FINE
      5:   ENDFOR
      6: L  P[4] 100mm/sec CNT100
     [End]

   [ INST ]                      [EDCMD]  >
```

R[1] changes as follows.

Loop	1st	2nd	3rd
1:	1	-	-
2:	1	2	3
3:	1	2	3
4:	1	2	3
5:	1->2	2->3	3
6:	-	-	3

(Note) "-" means the line is not executed.

Case 2: R[2]=1

TO is specified and Initial value is equal to Target value. Therefore, the condition of FOR statement is satisfied.

However, as the value of Loop counter is equal to Target value, the condition of ENDFOR statement is not satisfied. Therefore, FOR/ENDFOR loop is executed only once.

```
PROGRAM
                                    1/7
     1:   FOR R[1]=1 TO R[2]
     2: L  P[1] 100mm/sec CNT100
     3: L  P[2] 100mm/sec CNT100
     4: L  P[3] 100mm/sec FINE
     5:   ENDFOR
     6: L  P[4] 100mm/sec CNT100
   [End]

   [ INST ]                    [EDCMD]  >
```

R[1] changes as follows.

Loop	1st
1:	1
2:	1
3:	1
4:	1
5:	1
6:	1

(Note) “-“ means the line is not executed.

Case 3: R[2]=0

TO is specified and Initial value is larger than Target value. Therefore, the condition of FOR statement is not satisfied.

The cursor moves to the next line of combined ENDFOR statement, that is, line 6 and FOR/ENDFOR loop is not executed.

```
PROGRAM
                                    1/7
     1:   FOR R[1]=1 TO R[2]
     2: L  P[1] 100mm/sec CNT100
     3: L  P[2] 100mm/sec CNT100
     4: L  P[3] 100mm/sec FINE
     5:   ENDFOR
     6: L  P[4] 100mm/sec CNT100
   [End]

   [ INST ]                    [EDCMD]  >
```

R[1] changes as follows.

Loop	1st
1:	1
2:	-
3:	-
4:	-
5:	-
6:	1

(Note) “-“ means the line is not executed.

Special examples are shown as follows.

Start within FOR/ENDFOR loop: R[1]=0

When the following program is executed from line 3 and R[1]=0, the condition of ENDFOR statement is satisfied. Therefore, FOR/ENDFOR loop is repeated 4 times (0 to 3).

```
PROGRAM
                                    1/7
     1:   FOR R[1]=1 TO 3
     2: L  P[1] 100mm/sec CNT100
     3: L  P[2] 100mm/sec CNT100
     4: L  P[3] 100mm/sec FINE
     5:   ENDFOR
     6: L  P[4] 100mm/sec CNT100
   [End]
   Start from line 3.

   [ INST ]                    [EDCMD]  >
```

R[1] changes as follows.

Loop	1st	2nd	3rd	4th
1:	-	-	-	-
2:	-	1	2	3
3:	0	1	2	3
4:	0	1	2	3
5:	0->1	1->2	2->3	3
6:	-	-	-	3

(Note) “-“ means the line is not executed.

Start within FOR/ENDFOR loop: R[1]=5

When the following program is executed from line 3 and R[1]=5, the condition of ENDFOR statement is not satisfied. Therefore, FOR/ENDFOR loop is not repeated.

```
PROGRAM
                                          3/7
      1:   FOR R[1]=1 TO 3
      2: L  P[1] 100mm/sec CNT100
      3: L  P[2] 100mm/sec CNT100
      4: L  P[3] 100mm/sec FINE
      5:   ENDFOR
      6: L  P[4] 100mm/sec CNT100
   [End]
   Start from line 3
   [ INST ]                          [EDCMD]  >
```

R[1] changes as follows.

Loop	1st
1:	-
2:	-
3:	5
4:	5
5:	5
6:	5

(Note) "-" means the line is not executed.

JMP/LBL statements exist within FOR/ENDFOR loop:

In the following program, JMP/LBL statements exist within the FOR/ENDFOR loop. The cursor moves from line 3 to line 7 by JMP statement. FOR/ENDFOR loop between line 1 and 4 is not executed. AS line 7 is also within the FOR/ENDFOR loop and the condition of ENDFOR statement in line 8 is satisfied, FOR/ENDFOR loop between line 5 and 8 is repeated.
Finally, the value of R[1] is equal to 1 and R[2] is equal to 2.

```
PROGRAM
                                          1/10
      1:   FOR R[1]=1 TO 10
      2:   R[2]=0
      3:   JMP LBL[1]
      4:   ENDFOR
      5:   FOR R[2]=1 TO 2
      6: L  P[2] 100mm/sec CNT100
      7:   LBL[1]
      8:   ENDFOR
      9: L  P[3] 100mm/sec CNT100
   [End]
   [ INST ]                          [EDCMD]  >
```

R[2] changes as follows.

Loop	1st	2nd	3rd
1:	*	-	-
2:	0	-	-
3:	0	-	-
4:	-	-	-
5:	-	-	-
6:	-	1	2
7:	0	1	2
8:	0->1	1->2	2
9:	-	-	2

(Note) "-" means the line is not executed.

NOTE

Please be careful when you use JMP/LBL within FOR/ENDFOR loop.
Before you use them, please consider how the loop works. Otherwise, it may cause fatal errors.

Loop counter value is modified within FOR/ENDFOR loop:

Loop counter can be modified within the FOR/ENDFOR loops.
In the following example, when DI[1]=ON, R[1] is changed to 11. As this value is larger than Target value, this FOR/ENDFOR loop is finished.

```
PROGRAM
                                             1/10
      1:   FOR R[1]=6 TO 10
      2: L  P[1] 100mm/sec CNT100
      3: L  P[2] 100mm/sec CNT100
      4: L  P[3] 100mm/sec CNT100
      5:   IF DI[1]=OFF,JMP LBL[1]
      6:   R[1]=11
      7:   LBL[1]
      8:   ENDFOR
      9: L  P[4] 100mm/sec CNT100
    [End]

   [ INST ]                             [EDCMD]  >
```

R[1] changes as follows.

Loop	1st
1:	6
2:	6
3:	6
4:	6
5:	6
6:	6->11
7:	11
8:	11
9:	11

(Note) "-" means the line is not executed.

> **NOTE**
> Please be careful when you modify Loop counter value within FOR/ENDFOR loop. Changing Loop counter value within FOR/ENDFOR loop may cause an abnormal behavior or infinite loops.
> Please be careful when you modify Target value within FOR/ENDFOR loop when register or argument register is specified for them. Changing the value within FOR/ENDFOR loop may cause an abnormal behavior.

Loop counter value is calculated within FOR/ENDFOR loop:

In the following example, FOR/ENDFOR loop counter is calculated. After execution, Loop counter value will be 11. Loop counter value is different from Target value. However, this result is correct.

```
PROGRAM
                                           1/8
      1:   FOR R[1]=6 TO 10
      2: L  P[1] 100mm/sec CNT100
      3: L  P[2] 100mm/sec CNT100
      4: L  P[3] 100mm/sec CNT100
      5:   R[1]=R[1]+1
      6:   ENDFOR
      7: L  P[4] 100mm/sec CNT100
    [End]

   [ INST ]                             [EDCMD]  >
```

R[1] changes as follows.

Loop	1st	2nd	3rd
1:	6	-	-
2:	6	8	10
3:	6	8	10
4:	6	8	10
5:	6->7	8->9	10->11
6:	7->8	9->10	11
7:	-	-	11

(Note) "-" means the line is not executed.

This is explained as follows.
Line 1, R[1] is 6 (Initial value).
Line 5, R[1] is 7.
Line 6, R[1] is 7 and this satisfied the ENDFOR condition. R[1] is incremented to be 8.
Line 5, R[1] is 9.
Line 6, R[1] is 9 and this satisfied the ENDFOR condition. R[1] is incremented to be 10.
Line 5, R[1] is 11.
Line 6, R[1] is 11 and this does not satisfy the ENDFOR condition. The cursor goes to next line.
Finally the value of R[1] become 11 by these processes.

NOTE

In some cases, calculating Loop counter causes an alarm. This problem can be avoided by using DIV just before FOR/ENDFOR statement or input the same value in DATA screen. Especially when you calculate with real value use division, please make sure to use DIV.

R[1] = R[1] DIV 1

4.18.6 Alarms of FOR/ENDFOR Statement

Alarms occur in the following conditions with FOR/ENDFOR function.

- Execute when the number of FOR statement is smaller than ENDFOR statement.
- Execute when the number of ENDFOR statement is smaller than FOR statement.
- Execute when there are over 10 nested loops.
- Other than integer is used for Initial value or Target value in FOR statement.
- Other than integer is used for the value of Loop counter or Target value in ENDFOR statement.

- Execute when the number of FOR statement is smaller than ENDFOR statement.
  Execute the following program. As FOR statement for ENDFOR statement in line 5 does not exist, the alarm “INTP-669 Need FOR for ENDFOR in line 5” occurs in execution.

```
PROGRAM
                                              1/7
       1:
       2: L  P[1] 100mm/sec CNT100
       3: L  P[2] 100mm/sec CNT100
       4: L  P[3] 100mm/sec CNT100
       5:   ENDFOR
       6: L  P[4] 100mm/sec CNT100
     [End]

   [ INST ]                          [EDCMD]  >
```

- Execute when the number of ENDFOR statement is smaller than FOR statement.
  Execute the following program. As ENDFOR statement for FOR statement in line 1 does not exist, the alarm “INTP-670 Need ENDFOR for FOR in line 1” occurs in execution.

```
PROGRAM
                                              1/7
       1:   FOR R[1]=1 TO 10
       2: L  P[1] 100mm/sec CNT100
       3: L  P[2] 100mm/sec CNT100
       4: L  P[3] 100mm/sec CNT100
       5:
       6: L  P[4] 100mm/sec CNT100
     [End]

   [ INST ]                          [EDCMD]  >
```

- Execute when there are over 10 nested loops.
  The alarm “INTP-671 Too many FOR” occurs in execution.

- Other than integer is used for Initial value or Target value in FOR statement.
- Other than integer is used for the value of Loop counter or Target value in ENDFOR statement.
  Execute the following program. As the value of the register for Target value in FOR statement in line 2 is not integer, the alarm “INTP-672 (program, 2) Value type is not integer” occurs when line 2 is executed. In other cases, calculating the Loop counter, Initial value or Target value may cause the same alarm.

```
PROGRAM
                                          1/7
      1:   R[2]=1.234
      2:   FOR R[1]=1 TO R[2]
      3: L  P[2] 100mm/sec CNT100
      4: L  P[3] 100mm/sec CNT100
      5:   ENDFOR
      6: L  P[4] 100mm/sec CNT100
    [End]

   [ INST ]                          [EDCMD]  >
```

NOTE

In some cases, calculating a register used for Loop counter or Target value causes the alarm above even though the result and the value displayed in DATA screen is integer. This problem can be avoided by using DIV as follows just before FOR/ENDFOR statement or input the same value in DATA screen. Especially when you calculate with real value use division, please make sure use DIV.

R[1] = R[1] DIV 1

4.19 MIXED LOGIC INSTRUCTION

Overview

The Mixed Logic Instruction allows the use of various operator and data combinations in assignment statements, relational statements, and wait command statements in TP programs. The mixed logic instruction supports the NOT operator "!" and parentheses "()".

Mixed Logic Instructions can be specified on the “Register” menu, “I/O” menu, “IF/SELECT” menu, and “WAIT” menu.

Mixed Logic Instructions must be specified in parentheses, as shown below.

- DO[1]=(DI[1] AND !DI[2])
- IF (DI[1]) JMP LBL[1]
- WAIT (DI[1])

If not enclosed in parentheses, they are executed in the same way as other operation commands.

The mixed logic instruction supports Boolean data type variable flags and markers. For detail of flag and marker, refer to the Section “9.11 BACKGROUND LOGIC”.

Data types

Mixed logic instructions can use the data types below.

Table 4.19 (a) Data types

Type	Value	Data
Numeric	Numeric values can be handled as data. Both integer and real numbers can be used.	Register, constant, GI/O, AI/O, position register element, argument, system variable
Boolean	Data can assume either ON or OFF values.	DI/O, RI/O, UI/O, SI/O, WI/O, ON, OFF, flag, marker

⚠ **CAUTION**
Position data and palletizing data cannot be used with mixed logic instructions.

Operators

Mixed logic instructions can use the operators below.

Table 4.19 (b) Arithmetic operators

Operator	Operation
+	Addition of the left side and the right side
-	Subtraction of the right side from the left side
*	Multiplication of the left side and the right side
/	Division of the left side by the right side
MOD	Remainder of the division of the left side by the right side
DIV	Integer part of the quotient of the division of the left side by the right side

- The arithmetic operators can be used with numeric data only. If an attempt is made to use arithmetic operators with Boolean data, "INTP-203 Variable type mismatch" is generated.
- The output data of an arithmetic operator is always of numeric type.

Table 4.19 (c) Logical operators

Operator	Operation
AND	Logical product of the left side and the right side
OR	Logical sum of the left side and the right side
!	Logical negation of the left side and the right side

- The logical operators can be used with Boolean data only. If an attempt is made to use logical operators with numeric data, "INTP-203 Variable Type Mismatch" is generated.
- The output data of a logical operator is always of Boolean type.

Table 4.19 (d) Relational operators

Operator	Operation
=	Returns an ON value if the left side is equal to the right side. Otherwise, returns an OFF value.
<>	Returns an ON value if the left side is not equal to the right side. Otherwise, returns an OFF value.
<	Returns an ON value if the left side is less than the right side. Otherwise, returns an OFF value.
>	Returns an ON value if the left side is greater than the right side. Otherwise, returns an OFF value.
<=	Returns an ON value if the left side is equal to or less than the right side. Otherwise, returns an OFF value.
>=	Returns an ON value if the left side is equal to or greater than the right side. Otherwise, returns an OFF value.

- "=" and "<>" can be used with both numeric type data and Boolean type data.
- "<", ">", "<=", and ">=" can be used with numeric data only. If an attempt is made to use them with Boolean data, the error "INTP-203 Variable Type Mismatch" occurs.

The table below indicates the priority of operators.

Table 4.19 (e) Relational operators

Priority	Operator
High	!
	*, /, DIV, MOD
	+, —
Middle	<, >, <=, >=
	=, <>
	AND
Low	OR

Expressions

Mixed logic instructions can be used with assignment statements, conditional branch commands, and wait commands.

Assignment statements

Mixed logic instruction assignment statement examples are given below.

```
R[1] = ((GI[1] + R[1]) * AI[1])
DO[1] = (DI[1] AND (GI[1] = GI[2]))
```

- The leftmost = is for an assignment statement. The other = signs are for relational statements. The result of the expression on the right side is assigned to the data on the left side.
- The output data of an arithmetic operator is always of numeric type.
- If the data on the left side is of Boolean type and the result of the expression on the right side is of numeric type, the data on the left side is OFF if the value on the right side is less than 1 and greater than -1 and is ON if the value on the right side is greater than 1 or less than -1. This operation is the same as that for an ordinary assignment statement.
- If the data on the left side is of numeric type and the result of the expression on the right side is of Boolean type, the data on the left side is 0 if the value on the right side is OFF and is 1 if the value on the right side is ON. This operation is the same as that for an ordinary assignment statement.
- If a real number is assigned to a GO, AO, or integer type system variable, the fractional part is truncated.
- "Pulse" cannot be specified with a mixed logic instruction. To specify "Pulse", an ordinary operation command must be used.
- Position data and palletizing data cannot be specified on the left or right side of a mixed logic instruction. To specify position data or palletizing data, an ordinary operation command must be used.
- The maximum number of items (data items or operators) that can be used in an assignment statement is about 20. The exact maximum number of items that can be used depends on the data type.

The data below can be specified on the left side of an assignment statement.

Table 4.19 (f) Assigning data

Type	Data
Boolean	DO, RO, UO, SO, WO, flag, marker
Numeric	Register, GO, AO, position register element, system variable

Conditional statements

The following shows examples of using mixed logic instructions with conditional branch instructions.

```
IF (R[1] = (GI[1] + R[1]) * AI[1]) JMP LBL[1]
IF (DI[1] AND (!DI[2] OR DI[3])) JMP LBL[1]
```

- A mixed logic instruction expression can be used in the conditional statement of a conditional branch command.

- The result of a conditional statement must be of Boolean type.
- If the result of a conditional statement is ON, the executable statement (JMP LBL, for example) of the conditional branch command is executed.
- If a mixed logic is used in a conditional statement, the statements below can be used as an executable portion of a conditional branch command.

  JMP LBL[ ]
  CALL
  MIXED LOGIC STATEMENT
  PULSE STATEMENT

- A mixed logic assignment statement and a pulse statement can be specified in the executable statement of a conditional branch command only if the conditional statement contains a mixed logic expression. See the examples below.

  IF (DI[1]), DO[1]=(On)
  IF (DI[2]), DO[1]=Pulse

- The maximum number of items (data items or operators) in a conditional statement is about 20. The exact maximum number of items depends on the data type.

Wait commands

The following shows an example of using mixed logic commands with a wait command.

```
WAIT (DI[1] AND (!DI[2] OR DI[3]))
```

- A mixed logic expression can be specified in the conditional statement of a wait command.
- The result of a conditional statement must be of Boolean type.
- A wait command waits until the result of the expression becomes ON.
- It is not possible to specify "On+", "Off-", or "ERR_NUM" with a mixed logic command. To specify these, it is necessary to use ordinary operation command.
- The maximum number of items (data items and operators) in the conditional statement of a wait command is about 20. The exact maximum number of items depends on the data type.

Adding mixed logic commands

Editing mixed logic commands is complicated than editing ordinary commands. The reason for this is that they can use various data types and operators and there can be various combinations of them.

The functions below are provided to facilitate editing.

- To start mixed logic command teaching, a statement containing parentheses must be selected first.
- On the item selection menu, the items available according to the location of the statement appear.
- If a combination of items is invalid (for example, there are adjacent operators), an empty item is automatically inserted, prompting the user to select an item.
- When an item is selected, any related items are automatically deleted. For example, if an operator is deleted, the subsequent operated item is deleted at the same time.
- If the cursor is on an item in a mixed logic expression, and the expression is invalid, an error message appears on the prompt line. The message below appears.

Table 4.19 (g) Mixed logic error messages

Error message	Explanation
Parentheses mismatch	The number of opening parentheses is not equal to the number of closing parentheses.
Invalid index	Invalid index number.
Variable type mismatch	The data type does not match the operator.
Invalid parameter name	Invalid system variable name.
Untaught element	There is an un-taught item (...).
Invalid motion group	A specific operation group with a PR[ ] cannot be used in a program.
Invalid item for output	The item on the left side of the assignment statement is invalid.
Invalid item for Mixed Logic	The item cannot be used in a mixed logic expression.
Syntax error	Invalid statement.

Method to add a mixed logic commands

- In order to add a mixed logic command, select (…) in the right side of register instruction or I/O instruction, or the condition of IF instruction, or the condition of WAIT instruction.
- In order to change a mixed logic instruction, press F4, [CHOICE] while the cursor is positioned on that item. Available items appear.
- In order to insert a mixed logic instruction, press F1, "INSERT". "..." is inserted before the cursor, and an item selection menu appears.
- In order to delete a mixed logic instruction, move the cursor to an item and press F4, [CHOICE], then press F2, "DELETE". If an operator is deleted, the subsequent data item is also deleted.
- Except on the left side of an assignment statement, in order to add or delete the NOT (!), press F5, (!) while the cursor is on a digital I/O item in a mixed logic expression. A negation operator (!) is added or deleted.
- If the right side of an assignment statement contains a mixed logic command, in order to change the left side of the assignment statement, move the cursor to an item on the left side and press F4, [CHOICE]. A menu containing the items that can be specified on the left side of the assignment statement appears.

4.20 DIAGNOSIS INSTRUCTION

Diagnosis instruction is the instruction which is required for “Robot Condition Analysis” in ROBOGUIDE DiagnosticsPRO. In case that DIAG_REC instruction is taught before the motion instruction, the data needed for Robot Condition Analysis is recorded. For detail, refer to the help screen of ROBOGUIDE DiagnostcsPRO.

```
DIAG_REC((argument1), (argument2), (argument3))
               |           |            |
               ├Constant   ├Constatnt   ├Constant
               ├R[ i ]     ├R[ i ]      ├R[ i ]
               └AR[ i ]    └AR[ i ]     └AR[ i ]
```

Fig. 4.20(a) DIAG_REC instruction

Table 4.20(a) Arguments of DIAG_REC instruction

Argument 1	Data number. Positive integer equal to or less than10 can be specified. Diagnosis instruction which has same data number overwrites the existing data.
Argument 2	Specification of the recording switch register. The register number of the recording switch is specified by this argument. In case that the value of the specified register is not 0, the data is recorded. In case that this argument is set to 0, the data is recorded at any time.
Argument 3	Amount of the data to record. Positive integer equal to or less than the buffer size ($DTRECP.$BUF_SIZE) can be specified.

DIAG_REC_SEC instruction specifies the amount of the data recorded in seconds. Argument 2 of DIAG_REC instruction is omitted, and the data is recorded at any execution. It specifies length of the data by record time in seconds, instead of number of data as in DIAG_REC instruction.

```
DIAG_REC_SEC((argument1), (argument2))
                      │            │
                      ├─Constant   ├─Constatnt
                      ├─R[ i ]     ├─R[ i ]
                      └─AR[ i ]    └─AR[ i ]
```

Fig. 4.20(b) DIAG_REC_SEC instruction

Table 4.20(b) Arguments of DIAG_REC_SEC instruction

Argument 1	Data number. Positive integer equal to or less than10 can be specified. Diagnosis instruction which has same data number overwrites the existing data.
Argument 2	Length of the data to record in seconds. “Max. time for detection” is set in Diagnostics Interface. When Argument 2 is longer than “Max. time for detection”, the data will be recorded only to the “Max. time for detection”.

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
- Perform arc welding
- Send output signals to the peripherals
- Receive input signals from the peripherals

Before programming, design the outline of a program. In the design, incorporate the most effective method for the robot to do the target work. This enables efficient programming and ensures that only the instructions appropriate for the purpose are used.

Instructions must be selected from menus displayed on the teach pendant during programming. To teach a target position to the robot, the robot must be moved to the target position by jog feed.

After you have finished creating the program, change the program if necessary. To change, add, delete, copy, find, or replace an instruction, select the desired item from the menu displayed on the teach pendant.

This chapter describes the following:

- Tips on effective programming
- Turning on the power and jog feed
- Creating a program
- Changing a program

See Chapter 4 for the configuration of a program and the program instructions.

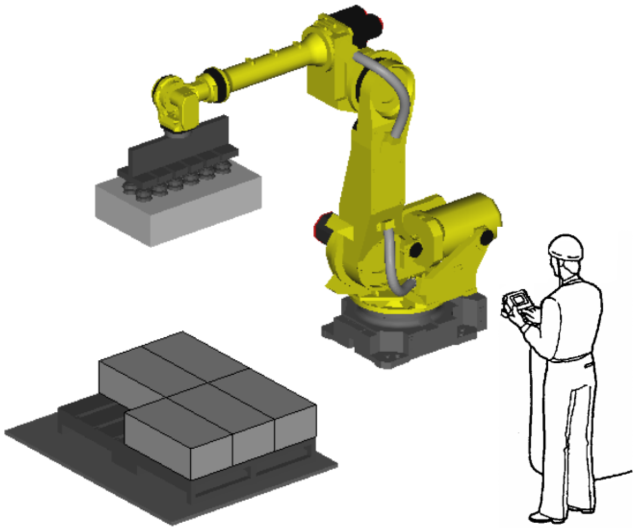


Fig. 5 Programming by teaching

5.1 TIPS ON EFFECTIVE PROGRAMMING

This section describes tips on effective programming. The following items are explained:

- Motion instructions
- Fixed positions

> **NOTE**
> This section describes tips on programming, but does not describe tips on jog feed.

5.1.1 Motion Instructions

Refer to the following instructions when teaching motions to the robot.

Workpiece hold position = FINE positioning

Use FINE positioning for all workpiece hold positions. The robot stops exactly at the workpiece hold position. When CNT positioning is used (explained next), the robot does not stop at taught points.

Moving around workpieces = CNT positioning

Use CNT positioning for moving around workpieces. The robot continuously moves to the next target point without stopping at taught points. If the robot moves near the workpieces, adjust the path of CNT positioning.

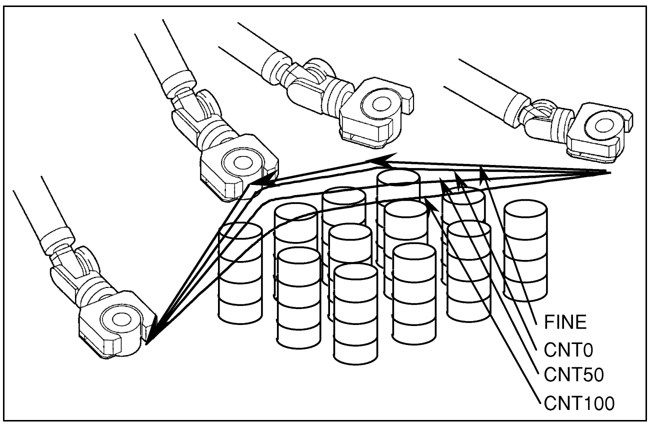


Fig. 5.1.1 (a) Adjusting the path of CNT positioning

Fixing the attitude of the tool

Cycle time is wasted when the robot motion considerably changes the attitude of the tool. The robot moves much faster when the attitude of the tool is changed smoothly and gradually. Teach positions so that the attitude of the tool changes as gradually as possible with respect to the robot.
When the attitude of the tool must be changed considerably, teach one large motion by dividing it into several small motions. Namely, teach positions so that the attitude of the tool changes gradually.

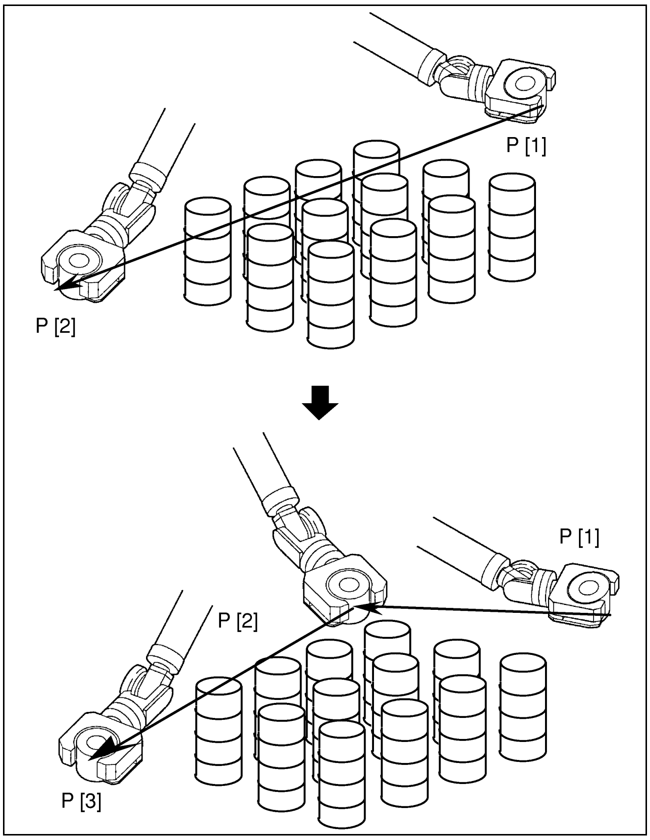


Fig. 5.1.1 (b) Teaching positions according to the tool attitude

To change the attitude of the hand as smoothly as possible:
1 Teach the first position of the work so that the robot has a normal attitude.
2 Move the robot to the last position of the work by jog feed. Then check that the robot has a normal attitude.
3 Teach the last position.
4 In accordance with the work, teach a position between the first and last positions.
5 Select a Cartesian coordinate system (World, user or jog coordinate system) and move the robot to the first position by jog feed.
6 Select the Cartesian coordinate system, move the robot toward the last position by jog feed, then stop the robot at the next position to be taught.
7 Correct the taught position so that the robot has a normal attitude.

> ⚠ **WARNING**
> If the J5 axis passes singular points (near 0 degrees) when the robot is operated by setting the move type to linear, the 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.

5.1.2 Predefined Position

The predefined position is the position that is referenced many times in a program. The predefined positions that are used often are the pounce position and the home (perch) position. You should define these positions to program efficiently or delete cycle time.

Pounce position

The pounce position is the reference position for all work. This is the safe position away from the motion area of the machine tool and peripheral device.

Home (perch) position

The home position, or perch position, is a safety position away from the machine tool and the workpiece transfer area. The reference position digital output signal is turned on when the robot is at this position. (See Section 3.10, ”SETTING A REFERENCE POSITION”.)

> **NOTE**
> HOME is a peripheral device I/O input signal, and does not represent a home position. A reference position is one of the home positions, but there is no utility used to move the robot to the reference position.

Other predefined position

The pounce position, reference position, or any other position can be defined as a predetermined position. Specify those positions that are frequently used in a program as predetermined positions.
When using the fixed position, use position registers (See Section 7.4) and macro instructions (See Section 9.1).

> ⚠ **CAUTION**
> If teaching is made by joint coordinates, changing the user coordinate system does not affect the position variables and position registers. However, note that both position variables and registers are affected by the user coordinate systems when the robot is taught in the Cartesian format.

> **NOTE**
> To move the robot to the same spatial position when the position register is shared by two programs, the two programs must have the same tool and user coordinate system.

5.2 TURNING ON THE POWER AND JOG FEED

5.2.1 Turning On the Power and Turning Off the Power

Turning on the power starts up the robot system. Turning on the power normally executes internal processing called a cold start or hot start, then the system is started up. The special operation is necessary to perform processing with a control or initial start. (See Section B.1, ”START MODE”.)

> ⚠ **CAUTION**
> Some systems require inspection before the robot is turned on. For the sake of safety, the system should be checked before the robot is turned on.

Hot start

You can select hot start if the hot start is setup when you start the robot system. The hot start is the function that saves the condition of the system just before power off and revives it after the next power on. (See Section 3.15, "SYSTEM CONFIG MENU".)

- If the hot start is set to FALSE ("Use HOT START" is set to FALSE on the system configuration screen), the system starts up with the cold start mode.
- If the hot start is set to TRUE ("Use HOT START" is set to TRUE on the system configuration screen), the system starts up in hot start mode.

HOT START done signal

You can set that the digital output signal (DO) is turned on when the hot start is finished. This function is set with the system configuration screen [6 SYSTEM. Config]. (See Section 3.15, "SYSTEM CONFIG MENU".)

Automatic start program

An automatic start program can be specified. The program is automatically started when the power is turned on. If override and parameter instructions are specified in the program to be started, the system can be customized when the power is turned on.

- In COLD START Autoexec program of the system configuration menu, register a program to be automatically started when the hot start is disabled. Such a program, if not defined, is not started.
- In HOT START Autoexec program of the system configuration menu, register a program to be automatically started when the hot start is enabled. 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 3.15, "SYSTEM CONFIG MENU".) And, the pulse command cannot be used in the automatic start program.

Program selection after power on

The condition of the program selection after the power on is the following:

- When hot start is disabled, it depends on the setting of "Restore selected program" on the system configuration menu.
  - TRUE : The program which had been selected at the power off is selected as it is.
  - FALSE : No program is selected.
- When hot start is effective, a program which had been selected at power off is selected as it is.

System condition

The table below lists settings in different start modes.

Table 5.2.1 System statuses in different start modes

	Hot start	
	Effective	**Disable (default setting)**
Contents of register	A	A
Override	A	C [10%]
Selection program	A	B (NOTE 3)
Execution line	A	C [First line]
Condition of I/O	A (NOTE 1)	C [All off]
TP screen	B (NOTE 2)	C [Hint screen]

A: All values that are current at power-down are saved and restored at power-up.
B: Only some of the values that are current at power-down are saved.
C: The values that are current at power-down are not saved. At power-up, the default values are set.

NOTE

1. Generally, the status existing at power-down is restored, but digital output (DO), being performed by a pulse instruction at power-down, is turned off. To restore the I/O status, specify the desired restoration status in [6 SYSTEM Config] (see Section 3.15, "SYSTEM CONFIG MENU").
   Even if the hot start is enabled, none of the output signals are resumed, but all output signals are turned off in the following cases:
   - The I/O allocation was changed before power-off.
   - The fuse of the I/O device blew, or the power to the I/O device was turned off.
   - The I/O device configuration was changed.
2. The screen type selected at power-down is restored, but the page, cursor, and other screen statuses are not restored. Instead, the screen is restored using the same page, cursor, and other screen statuses assumed immediately after a cold start.
3. The name of the main program that calls the subprogram is stored.

⚠ **CAUTION**

Before the power is turned on, system statuses in the corresponding start mode described above should be checked.

5.2.2 Three-Mode Switch

The three-mode switch is a key operation switch installed on the operator's panel or operation box. This switch is used to select an optimum robot operation mode according to the robot operation conditions and use status. There are operation modes AUTO, T1, and T2. See Fig. 5.2.2.

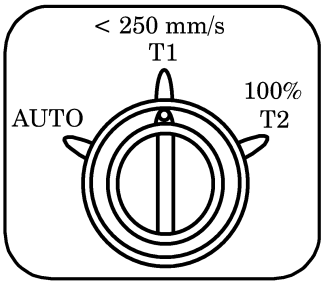


Fig. 5.2.2 Three-mode switch

When the three-mode switch is used to switch between operation modes, a message will be displayed on the screen of the teach pendant, and the robot halts. When the key is removed from the switch, the switch setting position can be fixed.

⚠ **CAUTION**

If switching between T1 or T2 mode and AUTO mode is made with the deadman switch kept held, a system error occurs. In this case, selected mode is not set until the deadman switch is released. Release the deadman switch, then hold the deadman switch again.

- Connection

Connect the safety fence signal to the safety fence. For connection of the safety fence signal, refer to the "FANUC Robot series R-30*i*B/R-30*i*B Plus CONTROLLER MAINTENANCE MANUAL" (B-83195EN) or the "FANUC Robot series R-30*i*B Mate/R-30*i*B Mate Plus CONTROLLER MAINTENANCE MANUAL" (B-83525EN) or the "FANUC Robot series R-30*i*B Mate

CONTROLLER Open Air MAINTENANCE MANUAL" (B-83555EN) or "FANUC Robot series R-30*i*B Compact Plus CONTROLLER MAINTENANCE MANUAL" (B-84035EN) or "FANUC Robot series R-30*i*B Mini Plus CONTROLLER MAINTENANCE MANUAL" (B-84175EN).

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 center point 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 center point 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 center point 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%.

- Safety fence

If you want to work with the safety fence kept open, it is necessary to set the three-mode switch to T1 or T2 before starting operating the robot.

- It is possible to operate the robot only when the teach pendant is enabled and the deadman switch is pressed (gripped).
- Disabling the teach pendant puts the robot in alarm condition, so the robot cannot run.
- When the teach pendant is enabled, but the deadman switch is not pressed, the robot is in alarm condition, so it cannot run.

> ⚠ **CAUTION**
> When checking the program you created, be sure to follow the "FANUC Robot SAFETY HANDBOOK" (B-80687EN).

- 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 [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.
The T2 mode is not the Manual high speed specified in ISO 10218-1. The jogging is available in T2 mode, but the speed at the tool center point and flange are both limited not to exceed 250 mm/sec for the Jogging.

- Program execution

A program can be executed only from the teach pendant.

- Robot speed at jogging

- The speeds at the tool center point and flange are both limited not to exceed 250 mm/sec.

- Robot speed at executing program

- The override value can be increased to up to 100%. There is no special speed limitation.
- When the switch is switched to T2 mode, the default override is limited to such a value that the speeds at the tool center point and flange are both equal to or less than 250mm/sec. The override can be changed with appropriate means such as the override key.
- While the switch is set in the T2 mode and the deadman switch on the teach pendant is released or strongly pressed, the override will be limited to such a value that the speeds at the tool center point and flange are both equal to or less than 250mm/sec. However if you use R-30iB or R-30iB Mate, this limitation is enable only in case of CE specification or NRTL specification.

- Safety fence

If you want to work with the safety fence kept open, it is necessary to set the three-mode switch to T1 or T2 before starting operating the robot.

- It is possible to operate the robot only when the teach pendant is enabled and the deadman switch is pressed (gripped).
- Disabling the teach pendant puts the robot in alarm condition, so the robot cannot run.
- When the teach pendant is enabled, but the deadman switch is not pressed, the robot is in alarm condition, so it cannot run.

> ⚠ **CAUTION**
> When checking the program you created, be sure to follow the "FANUC Robot SAFETY HANDBOOK" (B-80687EN).

- Fixing operation mode

When the switch is set in the T2 mode position, the operation mode can be fixed to T2 mode by removing the key.

- Troubleshooting

- When the switch is set in the T2 mode position, turning off the teach pendant enable switch stops the robot and causes an error message to appear. To release the error, set the teach pendant enable switch to on, then press the RESET key.

AUTO: Auto mode

The AUTO mode is intended for use at production.

- Program execution

A program can be executed from external devices and operator's panel. Program execution from the teach pendant is impossible if the switch is set in the AUTO mode position.

- Robot speed

The robot can be operated at a maximum speed.

- Robot speed at jogging

Jogging operation is not possible.

- Robot speed at executing program

- The robot can be operated at a maximum speed.

- Safety fence

Close the safety fence. When the safety fence is opened during program execution, the robot stops immediately (Please refer to "STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS for detail of stop type).

- Fixing operation mode

When the switch is set in the AUTO mode position, the operation mode can be fixed to AUTO mode by removing the key.

- Troubleshooting

- When the switch is set in the AUTO mode position, turning on the teach pendant enable switch stops the robot and causes an error message to appear. To release the error, set the teach pendant enable switch to off, then press the RESET key.

- Three-mode switch and program operation

The following table lists the relationships among the three-mode switch setting, safety fence status, teach pendant (TP) enabled/disabled, deadman switch setting and program-specified robot operation speed.

Relationships between three-mode switch settings and program operations

Three-Mode switch	Safety fence	TP enabled/ disabled	TP deadman	Robot status	Units that can be started	Program-specified operation speed
AUTO	Open	Enabled	Gripped	Servo off (fence open)		
			Released	Servo off (deadman, fence open)		
		Disabled	Gripped	Servo off (fence open)		
			Released	Servo off (fence open)		
	Closed	Enabled	Gripped	Alarm and stop (AUTO and TP disable)		
			Released	Servo off (deadman)		
		Disabled	Gripped	Operable	External start(*1)	Programmed speed
			Released	Operable	External start(*1)	Programmed speed
T1	Open	Enabled	Gripped	Operable	TP only	T1 speed
			Released	Servo off (deadman)		
		Disabled	Gripped	Alarm and stop (T1/T2 and TP disabled)		
			Released	Servo off (T1/T2 and TP disabled)		
	Closed	Enabled	Gripped	Operable	TP only	T1 speed
			Released	Servo off (deadman)		
		Disabled	Gripped	Alarm and stop (T1/T2 and TP disabled)		
			Released	Servo off (T1/T2 and TP disabled)		

Three-Mode switch	Safety fence	TP enabled/ disabled	TP deadman	Robot status	Units that can be started	Program-specified operation speed
T2	Open	Enabled	Gripped	Operable	TP only	Programmed speed
			Released	Servo off (deadman)		
		Disabled	Gripped	Alarm and stop (T1/T2 and TP disabled)		
			Released	Servo off (T1/T2 and TP disabled)		
	Closed	Enabled	Gripped	Operable	TP only	Programmed speed
			Released	Servo off (deadman)		
		Disabled	Gripped	Alarm and stop (T1/T2 and TP disabled)		
			Released	Servo off (T1/T2 and TP disabled)		

*1 External start
Remote mode: Program start on the line control panel
Local mode: Start button on the robot operation panel

> **NOTE**
> Please refer to "STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS for detail of the stop type. "Servo off" is performed the processing of Power-Off Stop or that of Controlled Stop. "Alarm and stop" is performed the processing of Hold.

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:

- Feed rate override: Robot motion speed (jog feed rate)
- Manual-feed coordinate system: Coordinate system for robot motion (jog feed type)

Feed rate override

A feed rate override is one of the two factors on which jog feed depends. The feed rate override is represented in percentage (%).
The current feed rate override is displayed at the upper right corner of the screen on the teach pendant. The value of the feed rate override can be changed by pressing the override key on the teach pendant.

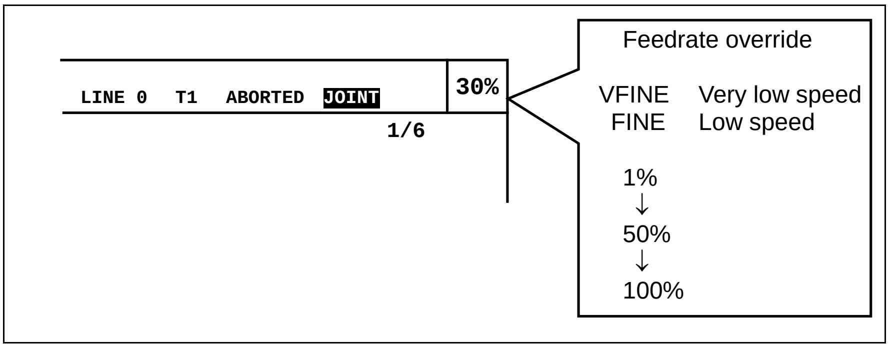


Fig. 5.2.3 (a) Screen display for feed rate override

Feed rate override 100% means that the robot moves at the maximum feed rate. The step feed-rate of FINE is specified by a system variable, $JOG_GROUP.$FINE_DIST in linear jog. (Standard: 0.1mm). In standard setting, each axis rotates at 0.001deg per step. The step width of VFINE is one-tenth of that of FINE.

NOTE
If VFINE or FINE is used as the current speed override, the robot makes a motion of a single step at a time. To resume the robot motion, release and press the jog key.

Table 5.2.3 (a) shows the change in feed rate override in standard setting when the override key is pressed.

Table 5.2.3 (a) Feed rate override (standard setting)

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 [SHIFT] key (*1)	VFINE → FINE → 5% → 50% → 100%

*1 Enabled only when $SHFTOV_ENB is 1

To change the feed rate override, press the override key. Whenever the override key is pressed while [SHIFT] key is pressed, the feed rate changes sequentially in the order: FINE, VFINE, 5%, 50%, and 100%. However, the feed rate is changed in this way only when system variable $SHFTOV_ENB = 1.

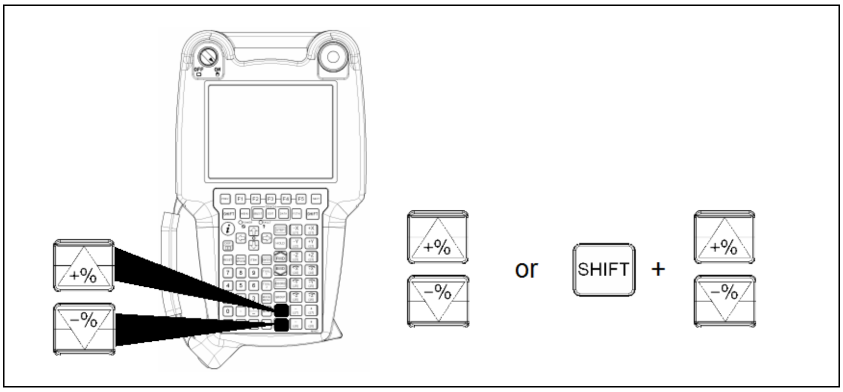


Fig. 5.2.3 (b) Override keys

A feed rate 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 feed rate override.

Customization of the change in feed rate override

The change in feed rate override when override key is pressed, or override key is pressed while [SHIFT] key is pressed can be customized. In order to customize the change in feed rate override, select the system variable $OVRD_SETUP on the system variables screen to display the detail screen of $OVRD_SETUP.

NOTE
Set the change in feed rate override when the override key is pressed while [SHIFT] key is pressed and the change in feed rate override when the override key is pressed without pressing [SHIFT] key independently.

```
SYSTEM Variables
     $OVRD_SETUP                        1/4
     1.$OVRD_NUM           0
     2.$OVERRIDE           [10] of INTEGER
     3.$OVRD_NUM_S         0
     4.$OVERRIDE_S         [10] of INTEGER

   [ TYPE ]
```

```
SYSTEM Variables
     $OVRD_SETUP.$OVERRIDE       1/10
       1   [1]            -2
       2   [2]            -2
       3   [3]            -2
       4   [4]            -2
       5   [5]            -2
       6   [6]            -2
       7   [7]            -2
       8   [8]            -2
       9   [9]            -2
      10   [10]           -2

   [ TYPE ]
```

- The value of the system variable $OVRD_SETUP.$OVRD_NUM is the number of the step of the change in feed rate override. (The value of the system variable $OVRD_SETUP.$OVRD_NUM_S is the number of the step of the change in feed rate override when the override key is pressed while [SHIFT] key is pressed.) In default setting, this system variable is set to 0, which means that this function is disabled. In order to increase or decrease the number of step of the change in feed rate override, change the value of this system variable. The setting value for this system variable must be equal to or greater than 1, and equal to or less than 10.
- The value of the system variable $OVRD_SETUP.$OVERRIDE ($OVRD_SETUP.$OVERRIDE_S) is the override value for each step. The setting value “-2” means that the override value for this step is not set. And the setting value “-1” means that the override for this step is VFINE. And the setting vale “0” means that the override vale for this step is FINE.
- Example
  In the following example, the change in feed rate override 8 steps when the override key is pressed without pressing [SHIFT] key. In this case, the system variable $OVRD_SETUP.$OVERRIDE[9] and [10] is not used.

  VFINE ↔ FINE↔10%↔20%↔30%↔50%↔80%↔100%

```
SYSTEM Variables
     $OVRD_SETUP                       1/4
     1.$OVRD_NUM           8
     2.$OVERRIDE           [10] of INTEGER
     3.$OVRD_NUM_S         0
     4.$OVERRIDE_S         [10] of INTEGER

   [ TYPE ]
```

```
SYSTEM Variables
     $OVRD_SETUP.$OVERRIDE       1/10
       1   [1]            -1
       2   [2]             0
       3   [3]            10
       4   [4]            20
       5   [5]            30
       6   [6]            50
       7   [7]            80
       8   [8]           100
       9   [9]            -2
      10   [10]           -2

   [ TYPE ]
```

> **NOTE**
> Set the system variables $OVRD_SETUP.$OVERRIDE[n] in ascending order of the step number. When the system variable $OVRD_SETUP.$OVRD_NUM is set to n, set from $OVRD_SETUP.$OVERRIDE[1] to $OVRD_SETUP.$OVRD_OVERRIDE[n]. In case that the setting is invalid, if the override key is pressed, the override value is always set to VFINE.

Jog feed rate

A jog feed rate is a speed at which the robot moves during jog feed. The jog feed rate is obtained by the following expression: If the following value exceeds the speed limit 250 mm/sec for the T1 or T2 mode described above, the operation speed is clamped at the one described earlier.

Jog feed rate (joint feed) (deg/sec, mm/sec) =

$$\text{Maximum joint feed rate} \times \frac{\text{Each axis jog override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Jog feed rate (linear feed) (mm/sec) =

$$\text{Maximum linear feed rate} \times \frac{\text{Jog override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Jog feed rate (Circular feed) (deg/sec) =

$$\text{Maximum circular feed rate} \times \frac{\text{Orientation Jog override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Each axis jog override	$SCR_GRP[g]. $JOGLIM_JNT[i] (%)
Jog override	$SCR. $JOGLIM (%)
Orientation jog override	$SCR. $JOGLIMROT (%)

Note: g is group number. i is axis number.

Fig. 5.2.3 (c) Jog feed rate

Manual-feed coordinate systems (Jog type)

Manual-feed coordinate systems determine how the robot moves during jog feed. The manual-feed coordinate systems are classified into three types:

- Joint jog (JOINT)

During joint jog, the robot moves independently around each axis according to each joint coordinate system. See Section 3.9 for the joint coordinate systems.

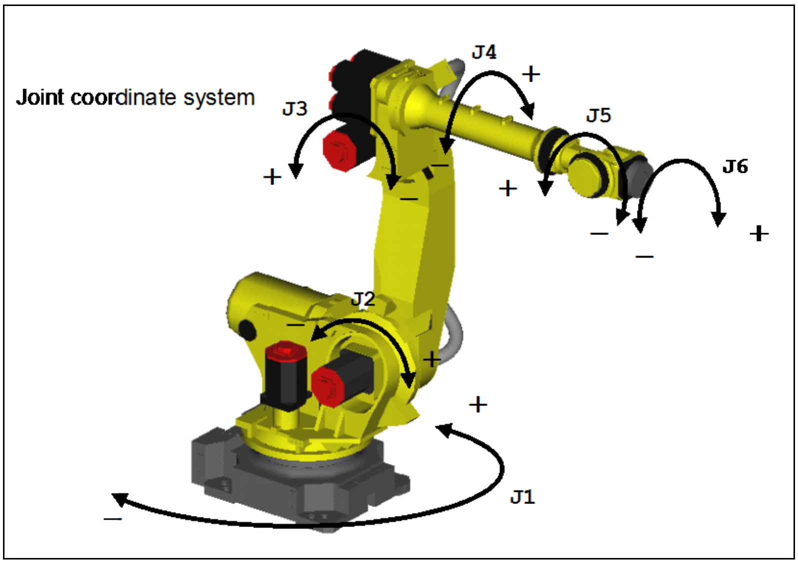


Fig. 5.2.3 (d) Joint jog

- Cartesian jog (XYZ)

During Cartesian jog, the tool center point of the robot moves along the X-, Y-, and Z-axes of the world or user or jog coordinate system. And the tool of the robot rotates around X-, Y-, and Z-axis of the world or user or jog coordinate system. (See Subsection 3.9.2, "Setting a User Coordinate System", and Subsection 3.9.3, "Setting a Jog Coordinate System".)

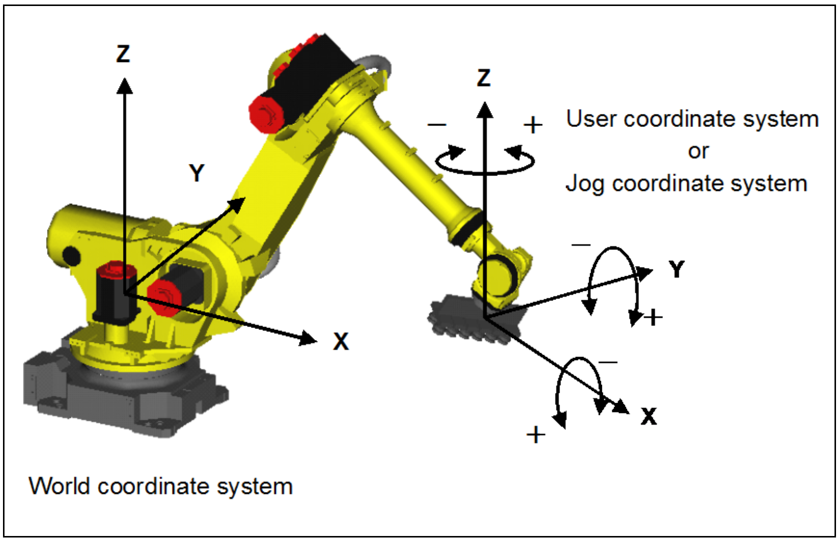


Fig. 5.2.3 (e) Cartesian jog

- Tool jog (TOOL)

During tool jog, the tool center point of the robot moves along the X-, Y-, and Z-axes of the tool coordinate system defined for the wrist of the robot. And the tool of the robot rotates around X-, Y-, and Z-axis of the tool coordinate system. (See Subsection 3.9.1, "Setting a Tool Coordinate".)

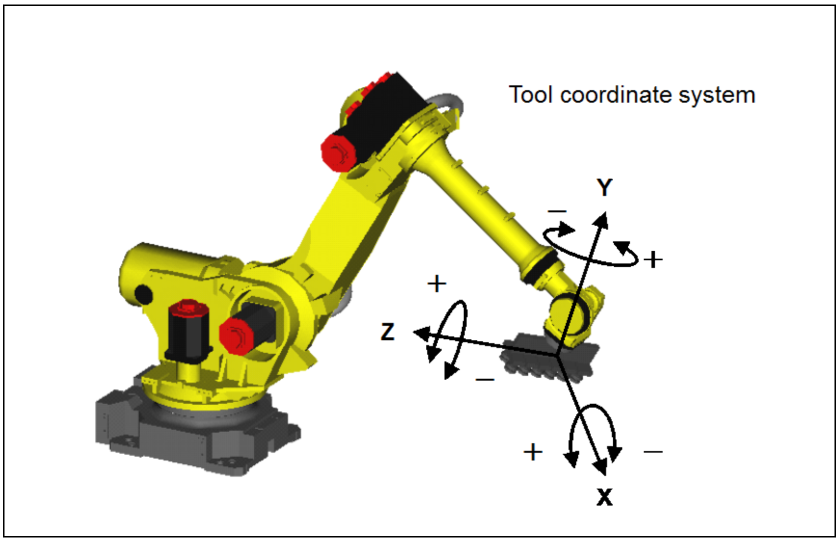


Fig. 5.2.3 (f) tool jog

Selecting a manual-feed coordinate system

The current manual-feed coordinate system is displayed in the status window of the teach pendant. Pressing [COORD] key displays a popup menu in reverse video at the upper right of the screen to call the user's attention. The popup menu in reverse video automatically disappears after a few seconds or when another key is pressed.

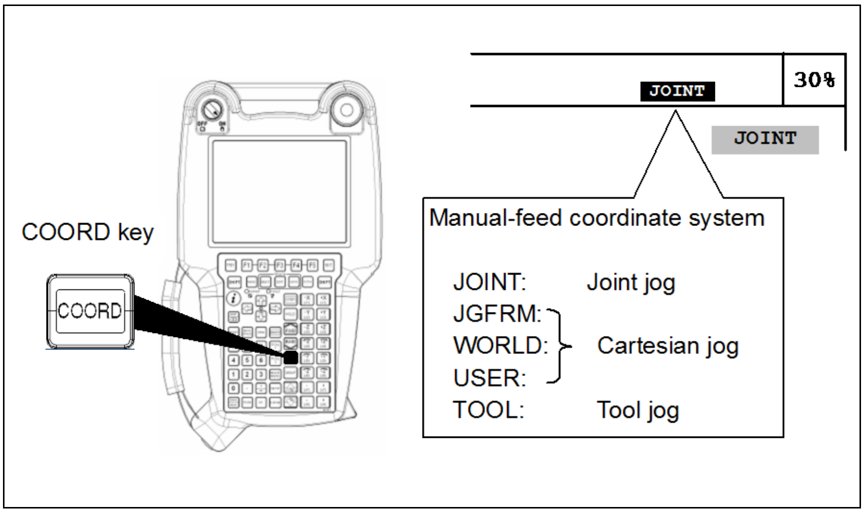


Fig. 5.2.3 (g) Screen display for manual-feed coordinate systems

Whenever [COORD] key on the teach pendant is pressed, the selected manual-feed coordinate system change cyclically.

Table 5.2.3 (b) Jog type selection sequence

Screen display	JOINT	→	JGFRM	→	WORLD	→	TOOL	→	USER	→	JOINT

When [COORD] key is pressed while [SHIFT] key is pressed, the ICON menu to change the manual-feed coordinate system is displayed at the bottom of the screen. By selecting one of the displayed ICONs, the manual-feed coordinate system can be changed. For detail about the ICON menu to change the manual-feed coordinate system, refer to the Subsection 2.3.1 Teach Pendant.

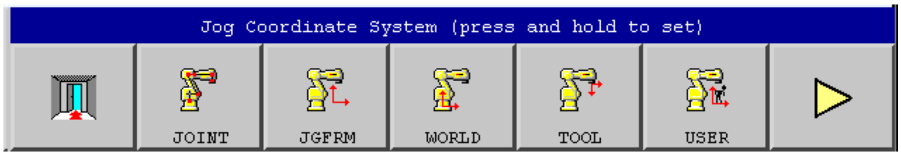


Fig. 5.2.3 (h) ICON menu to change the manual-feed coordinate system

> **NOTE**
> When [COORD] key is pressed while [SHIFT] key is pressed, the jog menu is displayed at the upper right corner of the screen at the same time as ICON menu to change the manual-feed coordinate system. About the jog menu, refer to Jog menu described later.

Enabling a wrist joint feed

In wrist joint feed, the attitude of the tool is not held during linear feed or rotational feed (Cartesian jog feed or 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.
  - The wrist axis is moved in axial movement while the position of the tool center point is held in rotational feed (attitude rotation about the wrist axis).

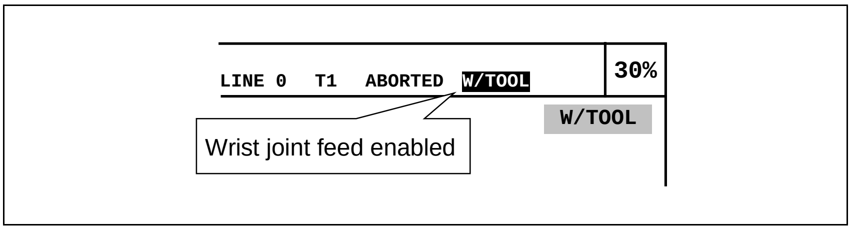


Fig. 5.2.3 (i) Indication that wrist joint feed is enabled

> **NOTE**
> When the motion instruction for linear or circular or circle arc motion under path control is executed, wrist joint feed has the same function as the wrist joint motion additional instruction (WRIST JOINT).

Jog feed of 5-axis robot

Since 5-axis robot cannot move under the full control of its Tool attitude, there are many attitudes which the robot cannot reach. For this reason, the jog feed of 5-axis robot has some unique characteristics.

1. During linear jog feed (along X-, Y-, Z-axis), 5-axis robot can exactly control its Tool attitude ONLY IF the flange surface faces in the vertical direction. If not, 5-axis robot moves in an alternative Tool attitude which is reachable. On the other hand, 5-axis robot can always exactly control its TCP position.
2. Rotational jog feed (around X-, Y-, z-axis) is automatically translated into the wrist joint feed.
3. During Joint jog feed, a message will be displayed and the robot temporarily stops. A message “Vertical fixture position” means that the flange surface now faces in the vertical direction. And a message “Horizontal fixture position” means that the flange surface now faces in the horizontal direction.

Switching to additional axes

In addition to the standard robot axes (usually 4 to 6 axes) in one operation group, up to three additional axes can be controlled as a subgroup.

> **NOTE**
> The user can switch to a subgroup by using the function menu or jog menu described below.

Additional axes can be jogged by using J7 and J8 jog keys without switching to the subgroup. But, because the assignment of J7 and J8 keys can be customized, the behavior by pressing J7 and J8 keys depends on the customized setting. For detail of J7 and J8 keys, refer to “Setting of J7, J8 keys” described later.

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.

```
UTILITIES Hints
                  Tool (.=10)  2
                  Jog          3
                  User         1
                  Group        2
                   [Robot]  Ext
```

Table 5.2.3 (c) Operation procedure using the jog menu

Operation	Procedure
Opening the menu	Press the manual feed [COORD] key while holding down [SHIFT] key.
Closing the menu	• Press the manual feed [COORD] key while holding down [SHIFT] key. • Press the [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 1 to 10 (Put the "." key to select 10.) • User coordinate system 0 to 9 • Jog coordinate system 1 to 5
Group switching (for a multi-group system only)	Numeric key (valid for existing group numbers only)
Subgroup switching (for a system with a subgroup)	After moving the cursor to the line containing Robot/Ext, switch between Robot and Ext by using the left/right cursor key. (The position of reverse video switches.)

⚠ **WARNING**

1 Be sure to remember the current coordinate system number/group number. Otherwise, in such a case, a robot may move in an unexpected direction at jog time, or a robot of an unexpected group may move, thus leading to a fatal accident.

2 After coordinate system number/group number switching, be sure to close the jog menu. If the jog menu is left open, the operator may change the coordinate system number or group number by touching a numeric key of the teach pendant unconsciously. In such a case, a robot may move in an unexpected direction at jog time, or a robot of an unexpected group may move, thus leading to a fatal accident.

Setting of J7 key and J8 key

Though J7 key and J8 key are used for the jog feed of the additional axes in the same group usually, if the setting is changed, J7 key and J8 key can be used to the jog feed for any axis. The setting of J7 and J8 keys is done in the "J7, J8 jog key Setup" in the system configuration menu. When the cursor is moved to the item "J7, J8 jog key Setup" and the [ENTER] key is pressed on the system configuration menu, the following screen will be displayed.

```
System/Config
      J7 J8 Jog keys                     1/6
        Group:       1         R-2000iB/165F

        1 J7 Group:   1         R-2000iB/165F
        2 J7 Axis:    7
        3 J7 Label:   ***************
        4 J8 Group:   2         Independent Axes
        5 J8 Axis:    1
        6 J8 Label:   G2 Servo Hand

   [ TYPE ]          GROUP
```

Fig 5.2.3 (j) J7, J8 Jog key setup screen

Table 5.2.3(d) J7, J8 Jog key setup

Item	Description
Group	The group number and the robot name whose setting is done currently are displayed. In order to setup in another group, press F3 GROUP to change the group.
J7 Group	This is the group number and the robot name of the axis that is jogged by pressing J7 key. The group number can be changed.

Item	Description
J7 Axis	This is the axis number that is jogged by pressing J7 key. The axis number can be changed. If this item is set to 0, J7 key is disabled.
J7 Label	This is the label name that is displayed beside “J7” in the current position screen. The label name can be changed within 15 characters. When this item is not set, or the null character is specified to this item, the group number and the axis number are displayed in the current position screen. (Example: G1 /J7).
J8 Group	This is the group number and the robot name of the axis that is jogged by pressing J8 key. The group number can be changed.
J8 Axis	This is the axis number that is jogged by pressing J8 key. The axis number can be changed. If this item is set to 0, J8 key is disabled.
J8 Label	This is the label name that is displayed beside “J8” in the current position screen. The label name can be changed within 15 characters. When this item is not set, or the null character is specified to this item, the group number and the axis number are displayed in the current position screen. (Example: G2 /J1).

The setting of J7 and J8 keys can be confirmed on any screen. When J7 key or J8 key is pressed without pressing [SHIFT] key, the pop up window is displayed at the upper right corner of the screen as follows. The name displayed in the pop up window can be changed by setting “J7 Label” or “J8 Label” on J7, J8 Jog Key Setup screen.

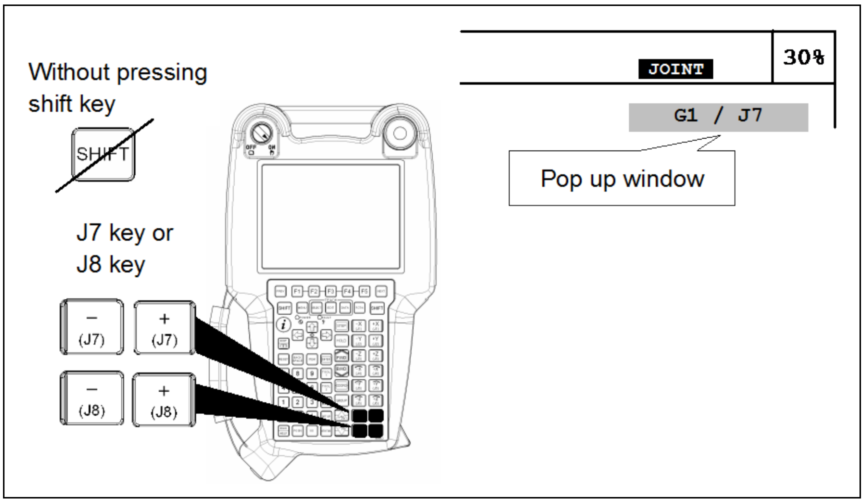


Fig 5.2.3 (k) Confirmation of the setting of J7 and J8 keys

NOTE

By default, J7 and J8 jog keys are assigned to the extended axes in the same group. But, in this initial condition, the current position and the label name of J7 and J8 are not displayed on the current position screen. And the pop up window for confirmation of the setting is also not displayed.

- If you want to enable these displays, go to “J7, J8 jog key setup” screen, then set “J7 Axis” or “J8 Axis” to 0, and then, set the proper value again. By this operation, the displays are enabled.
- If you want to disable these displays again, set “J7 Axis” or “J8 Axis” to 0, then set “J7 Group” or “J8 Group” to 0 (You will see the numbers automatically change back to the default value). By this operation, the displays are disabled.

Incremental Jog

With Incremental Jog function, the operator can jog the robot at an arbitrary step width per pressing the jog key. To enable this function, go to “System/Config” screen, enter detail screen of “Incremental Jog”, change “Enable function” “True”, and cycle power.

When this function is enabled, the step width currently selected is displayed under the screen display for feed rate override. In the case of following figure, the screan display means that the robot moves at 10mm per step in Linear jog and each axis rotates at 1 deg per step in Joint jog.

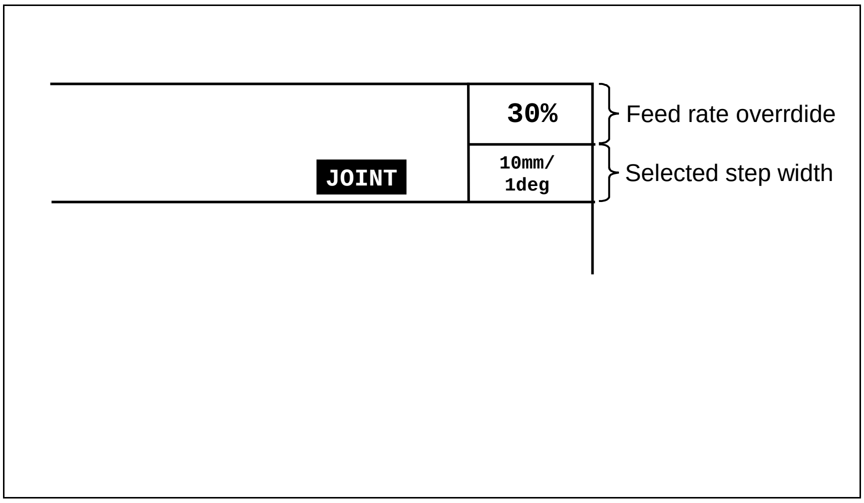


Fig 5.2.3 (l) Confirmation of the selected step width

To change the step width, please press the manual feed coordinate system key while holding down [SHIFT] key. The list of step width available are shown as follow. Please select desired step width.
When "Continuous" is selected, the behavior of jog feed is the same as when this function is disabled.

```
UTILITIES Hints
                    Tool (.=10)  1
                    Jog          1
                    User         0


                    Continuous
                    10mm/1deg
                    1mm/0.1deg
                    0.1mm/0.01deg
                    0.01mm/0.001deg
```

Step width can be changed in the detail screen of "Incremental Jog" in "System/Config" screen. To change the value of the step width, enter the value in "Amount of increments" in the range of 0.001 mm – 100 mm or 0.001 deg – 10 deg.

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.

> **NOTE**
> In case that any one of the following options is installed, the procedure of this operation differs.
> - J591 Robot operation without shit key function (Refer to Procedure 33-1 Moving the robot by jog feed (in case robot operation without [SHIFT] key function or jog operation without [SHIFT] key function is installed) in FANUC Robot series Optional Function OPERATOR'S MANUAL (B-83284EN-2).)
> - J739 Jog operation without shit key function (Refer to Procedure 33-1 Moving the robot by jog feed (in case robot operation without [SHIFT] key function or jog operation without [SHIFT] key function is installed) in FANUC Robot series Optional Function OPERATOR'S MANUAL (B-83284EN-2).)

> ⚠ **WARNING**
> 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 [COORD] key to select a desired manual-feed coordinate system on the teach pendant.

> **NOTE**
> When the manual-feed coordinate system is changed, the feed rate override is automatically changed to the safety value. (Default value 10%)

2 Press the override key to adjust the jog feed rate displayed on the teach pendant.
3 Hold the teach pendant and press the deadman switch on the back of the teach pendant. Continue pressing the deadman switch during jog feed.
4 Turn on the teach pendant enable switch.

> **NOTE**
> 1 If the deadman switch is released when the teach pendant enable switch is on, an alarm occurs. To reset the alarm, press and hold down the deadman switch again, then press the RESET key on the teach pendant.
> 2 If the operator is not accustomed to the operation of the robot or is not sure about the robot motions, low feed rate overrides should be set.

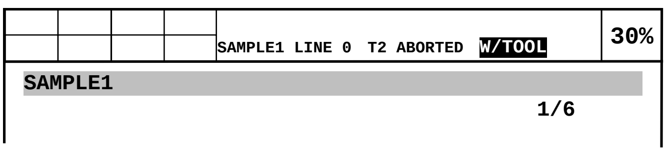
> ⚠ **WARNING**
> The robot starts its motion in the next step. If the jog feed of the robot needs to be stopped in an emergency in order to avoid danger, the operator should release the deadman switch or press the emergency stop button.

5 To move the robot by jog feed, press the jog key corresponding to the desired robot motion direction while pressing [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.

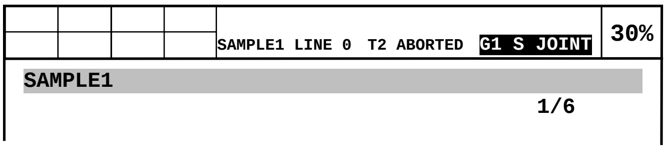
Switch to wrist joint feed

6 Press [FCTN] key. The function menu will be 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.



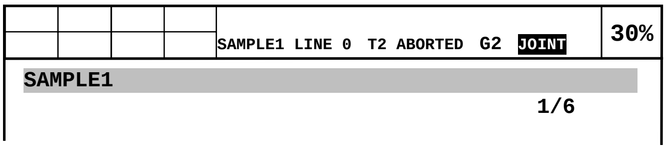
Switch to an extended axis

8 Press [FCTN] key. The function menu will be 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.
10 To terminate jog feed, turn off the teach pendant enable switch and release the deadman switch.



Switch jog group

In case that the controller supports the multi group, the axes in the selected jog group can be jogged. The selected jog group number is displayed in the status window as “Gn” (n is group number).



The jog group can be switched by the following 4 methods.

Switch in toggle type

1 Select “TOGGLE SUB GROUP” on function menu.
The jog group is switched step by step like G1, G2, G3, …, G1.

2 Press GROUP key.
The jog group is switched step by step like G1, G2, G3, …, G1. However, if the current jog group has the sub group, the jog group is switched to the sub group.

Direct switch

3 Set the value to the item Group in jog menu.
The jog group number is switched to the entered number.

4 Press the numeric key while the GROUP key is pressed.
The jog group number is switched to the number which specified by the pressed numeric key.

In the switch in toggle type, the jog group can be switched to only the motion group specified in the selected program by setting. Refer to “Setup for changing jog group according to the motion group of selected program” in Appendix C System variables.

5.3 CREATING A PROGRAM

To create a program, use the following procedure:

- Register a program and specify program information
- Modify standard instructions
- Teach motion instructions
- Teach various control instructions including spot instructions or arc welding instructions or sealing instructions or palletizing instruction

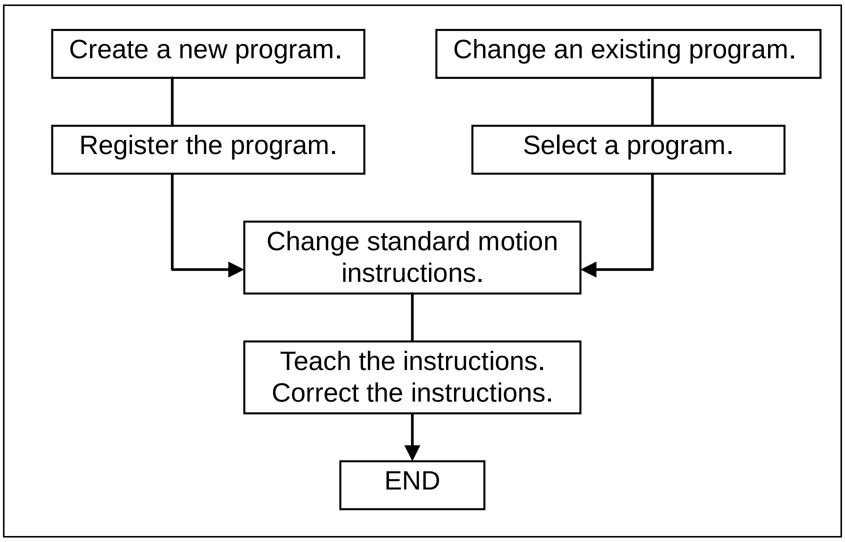


Fig. 5.3 Creating and changing a program

Registering a program
Create a null program with a new name.

Specifying program information
Specify the attributes of the program.

Changing standard motion instructions
Realign the standard instructions to be used when teaching motion instructions.

Teaching motion instructions
Teach a motion instruction and a supplementary motion instruction.

Teaching control instructions
Teach control instructions including a palletizing instruction.

Use the teach pendant to create a new program and correct an existing program. To do this, the teach pendant must be enabled beforehand.
To enable the teach pendant, satisfy the following condition:

■ The teach pendant enable switch must be turned on.

To prevent the program from being started by mistake, prohibit starting a program with a teach pendant while teaching. (See Fig. 2.3.1 (h), ”Function menu”.)

5.3.1 Registering a Program

Enter a program name and register the program. A program name consists of up to 36 alphanumeric characters including symbols to discriminate program names from one another. For the program name, see Subsection 4.1.1.
Register a program on the program registration screen.

> ⚠ **CAUTION**
> When a new program is made, the current program is halted.

Entering a program name

There are three methods for entering a program name:

- Words: Up to five words consisting of up to seven characters can be used as program names. Enter these reserved words, such as RSR, PNS, STYLE, JOB, and TEST, in $PGINP_WORD[1 TO 5] in advance (See Section 3.15, "SYSTEM CONFIG MENU").
- Uppercase or lowercase alphabetic characters: Any letter of the alphabet can be specified for a program name. The alphabetic characters combined with any numeric characters and/or any symbols are used as the characters of a program name.

> ⚠ **CAUTION**
> Asterisks (*) and at marks (@) should not be used in a program name.

Options

During optional settings, an overwrite or insert mode can be specified for character entry, or character string deletion.

- In the overwrite mode, entered characters are written over existing characters.
- In the insert mode, entered characters are inserted before the character pointed to by the cursor. In this case, all the characters to the right of the entered character(s) are shifted to the right.
- 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 : Change program name.
- Subtype Change subtype of program.
- Comments : Comments can be written in a program. Up to 16 alphanumeric characters and symbols, which can be used for a program name.
  In some cases, comments may not be entered.
- Group mask : Specifies a motion group to be controlled in a program.
- Write protection : Prevents a program from being changed.
- Interruption disable : Causes the program having no motion not to be paused by an alarm with a severity of WARN, PAUSE, STOP and SERVO, the emergency stop, and HOLD. However, this setting is not applied to the alarm that is generated by the program. In this case the program is stopped.
- Stack size : Specifies the memory size used at execution of the program call.

Procedure 5-2 Registering a program

Condition

- ■ The teach pendant must be enabled.

Step

1 Press [MENU] key to display the screen menu.
2 Select SELECT.
Alternatively, the following program list screen can also be displayed by pressing [SELECT] key.

```
Select
                 61276 bytes free      1/10
      No. Program name         Comment
       1  -BCKEDT-        [                 ]
       2  GETDATA     MR [Get PC Data       ]
       3  REQMENU     MR [Request PC Menu   ]
       4  SENDDATA    MR [Send PC Data      ]
       5  SENDEVNT    MR [Send PC Event     ]
       6  SENDSYSV    MR [Send PC Sysvar    ]
       7  SAMPLE1        [SAMPLE PROGRAM1   ]
       8  SAMPLE2        [SAMPLE PROGRAM2   ]
       9  PROG001        [PROGRAM001        ]
      10  PROG002        [PROGRAM002        ]

   [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR ]  >
     COPY    DETAIL   LOAD   SAVE AS   PRINT   >
```

3 Press the F2, CREATE key. The program registration screen is displayed.

```
    ---  Create Teach Pendant Program  ---

    Program Name:

                              --  End  --

                                     Alpha input 1
                                       Words
                                       Upper Case
                                       Lower Case
   Enter program name                  Options

      RSR     PNS    STYLE    JOB    TEST
```

4 Select a method for entering a program name (words or alphabetic characters) using the cursor keys.

```
    ---  Create Teach Pendant Program  ---

    Program Name:

                              --  End  --

                                     Alpha input 1
                                       Words
                                       Upper Case
                                       Lower Case
   Enter program name                  Options

    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 right arrow key to move the cursor to the right one character. Repeat this procedure until the program name is completely entered.

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.

```
   ---  Create Teach Pendant Program  ---

   Program Name:
    SAMPLE3
                                --  End  --




  Select function
|    |        | DETAIL |        |        |        |   |
```

7 To edit the registered program, press the F3, EDIT or the [ENTER] key. The program edit screen for the registered program will be displayed.

```
SAMPLE3
                                           1/1
     [End]





|    | POINT  |        |        |        | TOUCHUP | > |
```

8 To enter program information, press the F2, DETAIL key (or the [ENTER] key) on the screen of step 6. The program information screen is displayed.

```
Program detail
                                               1/7
      Creation Date:          16-Jan-1994
      Modification Date:      08-Mar-1994
      Copy Source:

      Positions: FALSE  Size:      312 Byte

      Program name:
       1 SAMPLE3
       2  Sub Type:      [None              ]
       3  Comment:       [SAMPLE PROGRAM 3 ]
       4  Group Mask:    [1,*,*,*,*,*,*,*  ]
       5  Write protect: [OFF               ]
       6  Ignore pause:  [OFF               ]
       7  Stack size:    [     500          ]

   END      DISP     NEXT
```

9 Specify the following program information items:

- To change a program name, move the cursor to the setting field, change the program name, then press the [ENTER] key.
- To change a subtype (see Subsection 4.1.3), press the F4, [CHOICE] key to display a subtype menu. Then, select None, Job, Process, Macro or Cond. JOB or PROCESS can be selected only when system variable $JOBPROC_ENB is set to 1.
- To enter comments, move the cursor to the setting field, enter the comments, then press the [ENTER] key (see Subsection 4.1.2).
- To specify a group mask, move the cursor to the setting field and select 1, *. The specified motion group is controlled (see Subsection 4.1.4). For safety, specify (*, *, *, *, *, *, *, *) for programs which do not contain any motion instructions. It is possible to change the initial setting at program creation. (Refer to Initial setting of the motion group in Appendix C System variables.)

> ⚠ **CAUTION**
> You cannot change the motion group of a program that contain operation instructions.

> **NOTE**
> If the system used does not have the multi-group setting, only either of the following settings is allowed: The first group is set as 1; An asterisk (*) indicating no group is set.

- To specify write protection, move the cursor to the setting field and select ON or OFF (see Subsection 4.1.5).
- To specify interruption disable, move the cursor to the setting field and press the function key (ON or OFF) (see Subsection 4.1.6). Select ON for programs not to be halted when an alarm occurs such as macro instructions or automatic start programs.
- To specify stack size, move the cursor to the setting field and press the [ENTER] key (see Subsection 4.1.7).

> **NOTE**
> To return to the list screen, press the [PREV] key repeatedly until the list screen is displayed.

10 After entering the program information items, press the F1, END key. The program edit screen for the registered program is displayed.

```
SAMPLE3
                                          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 will be displayed. Press the F key again. The screen for editing the standard operation statements will be displayed.

```
 | POINT |       |       |       | TOUCHUP | > |
```

- Press F1 POINT to list the standard operation statements.

Procedure 5-3 Changing a standard motion instruction

Condition

- ■ The program edit screen must be selected.
- ■ The teach pendant must be enabled.

```
SAMPLE3
                                          1/1
   [End]

 | POINT |       |       |       | TOUCHUP | > |
```

Step

1 Press the F1, POINT key. The standard motion instruction menu will be displayed.

```
SAMPLE3
                                        1/1
     [End]
       Default Motion  1
 1 J P[] 100% FINE
 2 J P[] 100% CNT100
 3 L P[] 100mm/sec FINE
 4 L P[] 100mm/sec CNT100
```

	ED_DEF				TOUCHUP	>

> **NOTE**
> If the instructions listed on the menu are necessary, they need not be changed.

2 To change a standard motion instruction, press the F1 ED_DEF.

```
Default Motion
                                        1/4
        1: J P[] 100% FINE
        2: J P[] 100% CNT100
        3: L P[] 100mm/sec FINE
        4: L P[] 100mm/sec CNT100
```

					DONE	

3 Move the cursor to the instruction item to be changed (motion type, feed rate, positioning type, or supplementary motion instruction) using the cursor keys.

```
Default Motion
                                        1/4
        1: J P[] 100% FINE
        2: J P[] 100% CNT100
        3: L P[] 100mm/sec FINE
        4: L P[] 100mm/sec CNT100




     Enter value
```

	REGISTER			[CHOICE]	DONE	

4 Select numeric keys and function keys to correct the instruction item.
To change the feed rate, for instance, move the cursor to feed rate. Enter a new value with numeric keys, then press the [ENTER] key.

```
Default Motion
                                          1/4
        1: J P[] 100% FINE
        2: J P[] 70% CNT100
        3: L P[] 100mm/sec FINE
        4: L P[] 100mm/sec CNT100




     Enter value
   REGISTER              [CHOICE]   DONE
```

5 When [CHOICE] is displayed in the F4 key name field, press the F4 key. Then, an option of another instruction item can be selected from the menu.

```
Default Motion
                                          1/4
        1:J P[] 100% FINE
  Motion modify 1 0% CNT100
 1 Fine          0mm/sec FINE
 2 Cnt           0mm/sec CNT100
 3
 4
 5
 6
 7
 8

                         [CHOICE]   DONE
```

```
Default Motion
                                          1/4
        1: J P[] 100% FINE
        2: J P[] 70% FINE
        3: L P[] 100mm/sec FINE
        4: L P[] 100mm/sec CNT100




                         [CHOICE]   DONE
```

6 Repeat steps 3 to 5 for each instruction to be changed.
7 After teaching is completed, press the F5, DONE key.

5.3.3 Teaching a Motion Instruction

A motion instruction moves the robot to the specified position in the work area at the specified feed rate 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, circle arc)
- Position variable: Stores data on positions to which the robot moves.
- Feed rate: Specifies the speed of the robot when it moves.
- Positioning type: Specifies whether positioning is performed at the specified position.
- Additional motion instruction: Specifies the instruction which executes with the robot motion.

Teaching a motion instruction is selected after a standard motion instruction is created. In this case, the current position (position data) is stored in the position variable.

- Press the F1 key to list the stored standard statements. Choose a desired statement from the list, and then program that statement.
- To program a single standard statement repeatedly, hold down [SHIFT] key and press the F1 key.

```
|      | POINT |      |      |      | 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].

```
SAMPLE3
                                        1/1
     [End]






|      | POINT |      |      |      | TOUCHUP | > |
```

3 Press the F1, POINT key to display the standard motion instruction menu.

```
SAMPLE3
                                       1/1
     [End]
 |       Default Motion  1  |
 |1 J P[] 100% FINE         |
 |2 J P[] 100% CNT100       |
 |3 L P[] 100mm/sec FINE    |
 |4 L P[] 100mm/sec CNT100  |




|      | 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.

```
SAMPLE3
                                             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 [SHIFT] key. This adds the previously specified motion instruction.

```
SAMPLE3
                                             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 Additional Motion Instruction

The additional motion instruction makes the robot do special work while it is moving according to the motion instruction. Some of the following additional motion instructions are provided (see Subsection 4.3.5 for the additional 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 additional motion instruction, place the cursor behind the motion instruction and press the F4, [CHOICE] key to display the additional motion instruction menu. Select an additional motion instruction from the menu. (See Appendix A.2 for the program instruction menu.)

```
Motion Modify    1
1 No option
2 Wrist Joint
3 ACC
4 Skip,LBL[]
5 BREAK
6 Offset/Frames
7 Offset,PR[ ]
8 --next page--
```

> **NOTE**
> The available additional motion instructions vary according to your software configuration.

Procedure 5-5 Teaching the additional motion instruction

Step

1 Place the cursor immediately behind the motion instruction.

```
SAMPLE3
                                        4/5

     4: L  P[3] 500mm/sec CNT10
   [End]

                            [CHOICE]
```

2 Press the F4, [CHOICE] key. The additional motion instruction menu will be displayed.

```
SAMPLE3
                                        4/5
                                   Motion Modify  3
                         Motion Modify  2 AFTER
      Motion Modify  1 remental    NCE BEFORE
      1 No option      l_Offset
      2 Wrist Joint    l_Offset,PR[
      3 ACC            ependent EV
      4 Skip,LBL[]     ultaneous EV
      5 BREAK          E BEFORE
      6 Offset/Frames  p,LBL,PR      t page--
      7 Offset,PR[ ]   ext page--
      8 --next page--

     4: L  P[3] 500mm/sec CNT10
                            [CHOICE]
```

3 Select a desired item.
For example, the following screen teaches an acceleration override instruction.

```
SAMPLE3
                                        4/5

     4: L  P[3] 500mm/sec CNT10 ACC100
   [End]


   Enter value
      REGISTER              [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 follows.
Press F4, [CHOICE]. The additional motion instruction menu will be displayed. Select “Incremental” on the additional motion instruction menu.

```
SAMPLE3
                                            4/5

      4: L  P[3] 500mm/sec CNT10
   [End]

                                   [CHOICE]
```

```
SAMPLE3
                                            4/5
                          Motion Modify  3
  Motion modi Motion modify  2 AFTER
  1 No option 1 Incremental    NCE BEFORE
  2 Wrist Join 2 Tool_Offset
  3 ACC        3 Tool_Offset,PR[
  4 Skip,LBL[] 4 Independent EV
  5 BREAK      5 Simultaneous EV
  6 Offset/Fra 6 TIME BEFORE
  7 Offset,PR[ 7 Skip,LBL,PR   t page--
  8 --next page-- 8 --next page--

      4: L  P[3] 500mm/sec CNT10
                                   [CHOICE]
```

```
SAMPLE3
                                            4/5

      4: L  P[3] 500mm/sec CNT10 INC
   [End]



   Position(P[3]) has been uninitialized.
                                   [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. Move the cursor to the position number and press F5, POSITION. The position data will be displayed.

```
SAMPLE3
                                          4/5

      4: L  P[3] 500mm/sec CNT10 INC
   [End]




   Enter value or press ENTER
                       [CHOICE] POSITION
```

```
SAMPLE3
      P[3] UF:0  UT:1          CONF:NDB 000
      X ******.***   mm   W ******.***  deg
      Y ******.***   mm   P ******.***  deg
      Z ******.***   mm   R ******.***  deg
    Position Detail



      4: L  P[3] 500mm/sec CNT10 INC
   [End]


   Enter value
                   CONF    DONE   [REPRE]
```

3 Enter the incremental amount directly.

```
SAMPLE3
      P[3] UF:0  UT:1          CONF:NDB 000
      X    500.000   mm   W      0.000  deg
      Y    100.000   mm   P      0.000  deg
      Z    100.000   mm   R      0.000  deg
    Position Detail



      4: L  P[3] 500mm/sec CNT10 INC
   [End]


   Enter value
                   CONF    DONE   [REPRE]
```

4 When you are fished to enter the position data, press F4, DONE.

```
SAMPLE3
                                         4/5

      4: L  P[3] 500mm/sec CNT10 INC
   [End]




   Enter value or press ENTER
                       [CHOICE] POSITION
```

5.3.5 Teaching a Control Instruction

A control instruction is a program instruction for the Robot controller that is not a motion instruction. The control instructions are as follows:

- Palletizing instruction
- Register instruction
- Position register instruction
- I/O (input/output) instruction
- Branch instruction
- Wait instruction
- Skip condition instruction
- Payload setting instruction
- Offset condition instruction
- Tool offset condition instruction
- Frame instruction
- Program control instruction
- Other instructions
- Multiaxis control instructions
- Motion group instruction
- FOR/ENDFOR instruction
- Diagnosis instruction
- Macro instruction

To teach a control instruction, first press the F1, [INST] key to display the menu. Then, select a desired control instruction item from the menu (see Appendix A.2 for the program instructions menu).

```
   Instruction  1        Instruction  2          Instruction  3
1 Registers           1 Miscellaneous         1 FOR/ENDFOR
2 I/O                 2 Skip                  2 Tool_Offset
3 IF/SELECT           3 Payload               3 LOCK PREG
4 WAIT                4 Offset/Frames         4 MONITOR/MON. END
5 JMP/LBL             5 Multiple control      5 String
6 CALL                6 Program control       6
7 Palletizing         7 MACRO                 7
8 --next page--       8 --next page--         8 --next page--
```

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
                                             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 will be displayed.

```
      PROGRAM  1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Palletizing
8 --next page--
```

3 To teach a register instruction, select Registers.
The following screens indicate that the value of register [1] is increased by one.

```
PROGRAM1
                                           2/2
        1: J  P[1] 100% FINE
  REGISTER statement 1
  1 ...=...
  2 ...=...+...
  3 ...=...-...
  4 ...=...*...
  5 ...=.../...
  6 ...=...DIV...
  7 ...=...MOD...
  8 ...=(...)

     Select item
                          [CHOICE]
```

```
PROGRAM1
                                           2/3
        1: J  P[1] 100% FINE
  REGISTER statement 1
  1 R[ ]
  2 PR[ ]
  3 PR[i,j]
  4 SR[ ]
  5
  6
  7
  8 --next page--

        2:   ...=...+...
                          [CHOICE]       >
```

```
PROGRAM1
                                  2/2
                               REGISTER statement 3
        1: J     REGISTER statement 2
  REGISTER statement 1
  1 R[ ]
  2 Constant
  3 DO[ ]
  4 DI[ ]
  5 RO[ ]
  6 RI[ ]
  7 GO[ ]                                 ge--
  8 --next page--          RFLOW
                           ge--

        2:   R[1]=...+...
                          [CHOICE]
```

```
PROGRAM1
                                        REGISTER statement 3
                         REGISTER statement 2
     REGISTER statement 1
     1 R[ ]
     2 Constant
     3 DO[ ]
     4 DI[ ]
     5 RO[ ]
     6 RI[ ]
     7 GO[ ]
     8 --next page--        RFLOW          ge--
                            ge--

        2:   R[1]=R[1]+...

                              [CHOICE]
```

```
PROGRAM1
                                               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 will be displayed.

```
      Instruction  1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Palletizing
8 --next page--
```

3 Select Registers.

```
REGISTER statement 1
1 ...=...
2 ...=...+...
3 ...=...-...
4 ...=...*...
5 ...=.../...
6 ...=...DIV...
7 ...=...MOD...
8 ...=(...)
```

4 Select PR[ ].
Teach the instruction assigning the Cartesian coordinates of the current position to the position register on the following screens.

```
REGISTER statement 1
1 R[ ]
2 PR[ ]
3 PR[i,j]
4 SR[ ]
5
6
7
8 --next page--
```

```
PROGRAM1
                                        2/3
      1: J  P[1] 100% FINE
   REGISTER statement 1
   1 Lpos
   2 Jpos
   3 UFRAME[ ]
   4 UTOOL[ ]
   5 P[ ]
   6 PR[ ]
   7
   8 --next page--

      2:   PR[1]=...

                           [CHOICE]
```

```
PROGRAM1
                                        3/3
      1: J  P[1] 100% FINE
      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 will be displayed.

```
  Instruction  1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Palletizing
8 --next page--
```

3 Select I/O.
Teach the instruction that turns on RO[1] on the following screens.

```
PROGRAM1
                                          2/2
      1: J  I/O statement 2
    I/O statememnt 1  ]=AI[ ]
    1 DO[ ]=...       ]=(...)
    2 R[ ]=DI[ ]      ]=(...)
    3 RO[ ]=...
    4 R[ ]=RI[ ]
    5 GO[ ]=...
    6 R[ ]=GI[ ]
    7 AO[ ]= ...     ext page--
    8--next page--

     Select item
                        [CHOICE]
```

```
PROGRAM1
                                          2/3
      1: J  P[1] 100% FINE
    I/O statement 1 ...
    1 On
    2 Off
    3 Pulse (,width)
    4 R[ ]
    5 (...)
    6
    7
    8

      2:   RO[1]=...
                        [CHOICE]
```

```
PROGRAM1
                                          3/3
      1: J  P[1] 100% FINE
      2:   RO[1]=ON
     [End]







                        [CHOICE]
```

For details of the instruction, see Chapter 4.

Procedure 5-10 Teaching a program call instruction

Step

1 Move the cursor to [End].
2 Press the F1, [INST] key. Then, the control instruction menu will be displayed.

```
   Instruction   1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Palletizing
8 --next page--
```

3 Select CALL. The following menu will be displayed.

```
CALL statement  1
1 CALL program
2 END
3
4
5
6
7
8
```

4 Select CALL program. The program list will be displayed.

```
PROGRAM1
                                    1/2
   SAMPLE1
   SAMPLE2
   SAMPLE3
   PROGRAM1
   PROGRAM2

 | PROGRAM | MACRO |  | INDIRECT | STRING | >
```

5 Select the called program on the program list.
- In order to call macro program, press F2, MACRO. The macro list is displayed. (If F1, PROGRAM is pressed after that, the program list is displayed again.)
- In order to specify the called program name indirectly by using the string register, press F4, INDIRECT.
- In order to enter a character string of the program name directly, press F5, STRING and enter the program name.

```
PROGRAMM1
                                          1/2
      1:   CALL PROGRAM2
   [End]

                          [CHOICE]
```

When the cursor is on the sub program name of CALL instruction, if the [ENTER] key is pressed, the edit screen is changed to the edit screen for the sub program. The sub program is displayed on the edit screen. If the [PREV] key is pressed, the edit screen is returned to the edit screen for the original program. It is possible that the next sub program which is called in the sub program is displayed by pressing the [ENTER] key. Up to 5 times, you can nest the sub program and return to the previous program.

```
PROGRAM1                                    PROGRAM2
                        1/2                                         1/5
     1:   CALL PROGRAM2 ------------>          1: J  P[1] 100% FINE
   [End]               ENTE                    2: J  P[2] 70% CNT50
                                               3: L  P[3] 500mm/sec CNT10
                       <----                   4: L  P[4] 500mm/sec CNT10
                       PREV                 [End]

              [CHOICE]                       POINT                 TOUCHUP  >
```

6 In order to specify the argument, move the cursor after the program name or the macro name, and press F4, [CHOICE]. The menu to select the argument type will be displayed.

```
Parameter select 1
1 R[ ]
2 Constant
3 String
4 AR[ ]
5 <None>
6 <Insert>
7 SR[ ]
8
```

7 Select the argument type to specify. Then, set some items to teach the argument for the CALL instruction in response to the argument type. For detail about the teaching of argument, refer to the Subsection 4.7.6 Arguments.

```
PROGRAMM1
                                        1/2
    1:   CALL PROGRAM2(R[1])
   [End]




                                 [CHOICE]
```

For details of the instruction, see Chapter 4.

Procedure 5-11 Teaching a macro instruction

Step

1 Move the cursor to [End].
2 Press the F1, [INST] key. Then, the control instruction menu will be displayed.

```
   Instruction    2
1 Miscellaneous
2 Skip
3 Payload
4 Offset/Frames
5 Multiple control
6 Program control
7 MACRO
8 --next page--
```

3 Select MACRO. The macro name list will be displayed.

```
PROGRAM1
                                        1/2
     Get Data
     Request Menu
     Send data
     Send Event
     Send Sysvar
     Hand Open
     Hand Close
```

4 Select the macro name on the macro name list.

```
PROGRAM1
                                      1/2
     1:   Hand Open
  [End]

            [CHOICE]
```

5 In order to specify the argument, move the cursor after the macro name, and press F4, [CHOICE]. The menu to select the argument type will be displayed.

```
Parameter select 1
1 R[ ]
2 Constant
3 String
4 AR[ ]
5 <None>
6 <Insert>
7 SR[ ]
8
```

6 Select the argument type to specify. Then, set some items to teach the argument for the macro instruction in response to the argument type. For detail about the teaching of argument, refer to the Subsection 4.7.6 Arguments.

```
PROGRAM1
                                      1/2
     1:   Hand Open(R[1])
  [End]

            [CHOICE]
```

For details of the instruction, see Chapter 4 and Macro Instruction (Section 9.1).

Procedure 5-12 Teaching move group instructions

Step

1 Move the cursor to the line number of a desired move statement (other than for circular motion or circle arc motion).

```
PROGRAM1
                                           1/2
     1: L  P[1] 1000mm/sec CNT100
   [End]

 [ INST ]                        [EDCMD]  >
```

2 Press F1, [INST]. Then, a list of control instructions is displayed.

```
   Instruction   3
1 FOR/ENDFOR
2 Tool_Offset
3 LOCK PREG
4 MONITOR/MON. END
5 Independent GP
6 Simultaneous GP
7 String
8 --next page--
```

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
                                           2/2
      1:Independent GP
       :GP1 L  P[1] 1000mm/sec CNT100
       :GP2 L  P[1] 1000mm/sec CNT100
   [End]

 [ INST ]                        [EDCMD]  >
```

4 For a move statement within the move group instructions, edit the move type, move speed, and positioning type in the same way as for an ordinary move statement.
Note that the following operations cannot be performed:

- Changing the move type to circular (The move type can be changed to the circle arc, but the changed instruction cannot be executed.)
- Specification of position data type (R[], PR[])

- Position number change
- Teaching of additional motion instructions (Deletion is allowed.)
- Deletion/creation of move groups
- Position modification by SHIFT + TOUCHUP

For details of instructions, see Chapter 4.

Procedure 5-13 Teaching parameter instructions

Step

1 On the program edit screen, press F1, [INST] and select Miscellaneous on the control instruction menu. Then, select Parameter name on the menu displayed by selecting Miscellaneous.

```
PNS0001
                                       1/1
    [End]
  Miscellaneous statement 1
  1 $...=...
  2 ...=$...
  3
  4
  5
  6
  7
  8

   Select item
                         [CHOICE]
```

2 Select “2. …=$...”.

```
PNS0001
                                       1/2
       1: ...=$...
  Miscellaneous statement 1
  1 R[ ]
  2 PR[ ]
  3
  4
  5
  6
  7
  8

                         [CHOICE]
```

3 Select “1.R[ ]”, and enter the register number.

```
PNS0001
                                          1/2
     1:    R[1]=$...
  [End]








  Press ENTER
                           [CHOICE]
```

4 Press F4, [CHOICE]. The menu of the system variables will be displayed. If the [ENTER] key is pressed, the character string of the system variable name can be entered directly.
When F4, [CHOICE] is pressed, the following menu will be displayed.

```
PNS0001
                                          1/2
     1:    R[1]=$...
  Parameter menu 1
  1 DEFPULSE
  2 WAITTMOUT
  3 RCVTMOUT
  4 SHELL_WRK.$CUR_STYLE
  5 SHELL_WRK.$CUR_OPTA
  6 SHELL_WRK.$CUR_OPTB
  7 SHELL_WRK.$CUR_OPTC
  8 --next page--

                           [CHOICE]
```

Select “1. DEFPULSE”.

```
PNS0001
                                          2/2
     1:    R[1]=$DEFPULSE
  [End]










  [ INST ]                            [EDCMD]
```

When the [ENTER] key is pressed, a character string can be entered directly.

```
PNS0001
                                         1/2
      1:    R[1]=
       :    $
    [End]






                                       Alpha input 1
                                       1 Word
                                       2 Upper Case
                                       3 Lower Case
                                       4 Options
   $        [        ]        .
```

Enter the system variable name.

For details of instructions, see Chapter 4.

Procedure 5-14 Teaching FOR/ENDFOR instructions

Step

1 Insert lines in the beginning and the ending of the section which is executed repeatedly. In this example, insert blank lines in line 1 and 5.

```
PROGRAM
                                         1/7
       1:
       2: L  P[2] 100mm/sec CNT100
       3: L  P[3] 100mm/sec CNT100
       4: L  P[1] 100mm/sec FINE
       5:
       6: L  P[3] 100mm/sec CNT100
    [End]






 [ INST ]                        [EDCMD]  >
```

2 Move the cursor to line 1, and press F1, [INST].

3 Select FOR/ENDFOR.

```
PROGRAM
                                         1/7
       1:                 Instruction  3
               Instruc    1 FOR/ENDFOR
    Instruc    1 Miscella 2 Tool_Offset
 1 Register    2 Skip     3 LOCK PREG
 2 I/O         3 Payload  4 MONITOR/MON.END
 3 IF/SELEC    4 Offset/F 5 String
 4 WAIT        5 Multiple 6 DIAGNOSE
 5 JMP/LBL     control    7
 6 CALL        6 Program  8 --next page--
 7 Palletiz    7 MACRO
 8 --next p

 [ INST ]                        [EDCMD]  >
```

4 Select FOR. The FOR statement is taught.

```
PROGRAM
                                           1/7
       1:
  FOR statement 1  100mm/sec CNT100
  1 FOR            100mm/sec CNT100
  2 ENDFOR         100mm/sec FINE
  3
  4                100mm/sec CNT100
  5
  6
  7
  8

   [ INST ]                         [EDCMD]  >
```

5 The cursor automatically moves to the index of the register. Input the value of the index. In this example, input 1.

```
PROGRAM
                                           1/7
       1:   FOR R[...]=... TO ...
       2: L  P[2] 100mm/sec CNT100
       3: L  P[3] 100mm/sec CNT100
       4: L  P[1] 100mm/sec FINE
       5:
       6: L  P[3] 100mm/sec CNT100
    [End]

   [ INST ]                         [EDCMD]  >
```

6 The cursor automatically moves to Initial value part. In this example, Constant is selected and the value is 1. To select the register or the argument register, press F4, [CHOICE] and select one of them.

```
PROGRAM
                                           1/7
       1:   FOR R[1]=... TO ...
  FOR statement 1  100mm/sec CNT100
  1 Constant       100mm/sec CNT100
  2 R[ ]           100mm/sec FINE
  3 AR[ ]
  4                100mm/sec CNT100
  5
  6
  7
  8

   [ INST ]                         [EDCMD]  >
```

```
PROGRAM
                                            1/7
        1:    FOR R[1]=1 TO ...
        2: L  P[2] 100mm/sec CNT100
        3: L  P[3] 100mm/sec CNT100
        4: L  P[1] 100mm/sec FINE
        5:
        6: L  P[3] 100mm/sec CNT100
     [End]

   [ INST ]                          [EDCMD]  >
```

7 The cursor automatically moves to TO/DOWN part. In this example, press the [ENTER] key to select TO. To select DOWNTO, press F4, [CHOICE] and select DOWNTO.

```
PROGRAM
                                            1/7
        1:    FOR R[1]=1 TO ...
        2: L  P[2] 100mm/sec CNT100
        3: L  P[3] 100mm/sec CNT100
        4: L  P[1] 100mm/sec FINE
        5:
        6: L  P[3] 100mm/sec CNT100
     [End]

   [ INST ]                          [EDCMD]  >
```

```
PROGRAM
                                            1/7
        1:    FOR R[1]=1 TO ...
  FOR statement 1   100mm/sec CNT100
  1 TO              100mm/sec CNT100
  2 DOWNTO          100mm/sec FINE
  3
  4                 100mm/sec CNT100
  5
  6
  7
  8

   [ INST ]                          [EDCMD]  >
```

8 The cursor automatically moves to Target value part. In this example, select R[]. Input 2 for the index of the register.

```
PROGRAM
                                        1/7
        1:   FOR R[1]=1 TO ...
 FOR statement 1 | 100mm/sec CNT100
 1 Constant      | 100mm/sec CNT100
 2 R[ ]          | 100mm/sec FINE
 3 AR[ ]
 4               | 100mm/sec CNT100
 5
 6
 7
 8

   [ INST ]                     [EDCMD]  >
```

```
PROGRAM
                                         1/7
        1:   FOR R[1]=1 TO R[2]
        2: L  P[2] 100mm/sec CNT100
        3: L  P[3] 100mm/sec CNT100
        4: L  P[1] 100mm/sec FINE
        5:
        6: L  P[3] 100mm/sec CNT100
     [End]

   [ INST ]                     [EDCMD]  >
```

9 Move the cursor to line 5 and press F1, [INST].

10 Select FOR/ENDFOR.

```
PROGRAM
                                        5/7
        1:             Instruction  3
                Instruc 1 FOR/ENDFOR
    Instruc 1 Miscella 2 Tool_Offset
 1 Registe 2 Skip      3 LOCK PREG
 2 I/O     3 Payload   4 MONITOR/MON.
 3 IF/SELE 4 Offset/F  END
 4 WAIT    5 Multiple  5 String
 5 JMP/LBL control     6 DIAGNOSE
 6 CALL    6 Program   7
 7 Palleti 7 MACRO
 8 --next page

   [ INST ]                     [ EDCMD ]  >
```

11 Select ENDFOR. The ENDFOR statement will be taught.

```
PROGRAM
                                              5/7
        1:   FOR R[1]=1 TO R[2]
 FOR statement 1 | 100mm/sec CNT100
 1 FOR           | 100mm/sec CNT100
 2 ENDFOR        | 100mm/sec FINE
 3
 4               | 100mm/sec CNT100
 5
 6
 7
 8

    [ INST ]                          [EDCMD]   >
```

```
PROGRAM
                                              5/7
        1:   FOR R[1]=1 TO R[2]
        2: L  P[2] 100mm/sec CNT100
        3: L  P[3] 100mm/sec CNt100
        4: L  P[1] 100mm/sec FINE
        5:   ENDFOR
        6: L  P[3] 100mm/sec CNT100
      [End]

    [ INST ]                          [EDCMD]   >
```

For details of instructions, see Chapter 4.

5.3.6 TP Start Prohibition

The Robot controller can execute the program immediately while editing it. To prevent the program from being executed by mistake, you can prohibit starting the program while teaching with this function.
When you select Disable FWD/BWD in the function menu, starting a program with a teach pendant is prohibited. At this time, ”FBD” is reversibly 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 on 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 [SHIFT] key, and press [FWD] key or [BWD] key in prohibition mode. At this time, a warning message, ”Teach pendant is disabled”, is displayed in the status window.

Jog feed during TP start prohibition

A system variable can be set to enable jog feed only in the TP start prohibition state.
To make this setting, system variable $SCR.$TPMOTNENABL is used.
To enable this function (to enable jog feed only in the TP start prohibition state), change the value of system variable $SCR.$TPMOTNENABL from 0 to 1 (or from 2 to 3) on the system variable screen.
The table below indicates the relationship between the value of system variable $SCR.$TPMOTNENABL and whether TP start and jog feed are enabled.

Table 5.3.6 Setting for Jog feed during TP start prohibition

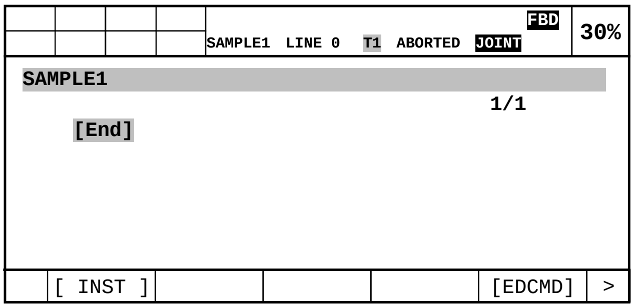
$SCR.$TPMOTNENABL	TP start	Jog feed
0	Enabled	Enabled
1	Enabled	Disabled
2	Disabled	Enabled
3	Disabled	Enabled

With the standard setting, this function is disabled (jog feed is enabled irrespective of whether the teach pendant can start a program).

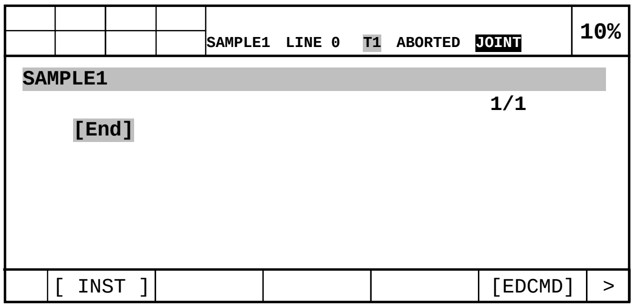
Procedure 5-15 Prohibiting Starting with Teach Pendant

Step

1. Press [FCTN] key. The function menu will be displayed.
2. Select 2 Disable FWD/BWD. ”FBD” is reversibly displayed in the uppermost right hand line of the screen.



3. To release the prohibition mode, select ”2 Disable FWD/BWD” on the function menu again. ”FBD” disappears and the override is reduced to a setting of $SCR.$FWDENBLOVRD.



Procedure 5-16 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.

```
SAMPLE1  LINE 0   T1  ABORTED  JOINT        30%
SAMPLE1
                                      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 lower than the setting of $SCR.$FWDENBLOVRD.

```
                                     FBD
SAMPLE1  LINE 0   T1  ABORTED  JOINT        10%
SAMPLE1
                                      1/1
    [End]

   [ INST ]                         [EDCMD]  >
```

5.4 CHANGING A PROGRAM

The method of changing the contents of an existing program is described on this section.

- Selecting a program
- 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 whether a comment in the program instruction is display or not
  - Restoring a program editing operation
  - Changing a program instruction to remark.

5.4.1 Selecting a Program

When selecting a program, call the registered program to display the program edit screen for editing, changing and executing a program.
Once a program is selected, the program is effective until another program is selected. While another screen is displayed such as the current position screen, the currently selected program is started by the start switch.

- In case that the teach pendant is enabled, when a program is selected, the current or halted program is forcibly terminated.
- In case that the teach pendant is disabled, while a program is being executed or halted, another program cannot be selected.

Select a program on the program selection screen.

Procedure 5-17 Selecting a program

Step

1 Press [MENU] key.
2 Select SELECT.
Alternatively, press the SELECT key to enable a program to be selected. In this case, the program selection screen will be displayed.

```
Select
                 61092 bytes free      9/11
      No. Program name         Comment
       1  -BCKEDT-       [                 ]
       2  GETDATA     MR [Get PC Data      ]
       3  REQMENU     MR [Request PC Menu  ]
       4  SENDDATA    MR [Send PC Data     ]
       5  SENDEVNT    MR [Send PC Event    ]
       6  SENDSYSV    MR [Send PC Sysvar   ]
       7  SAMPLE1        [SAMPLE PROGRAM 1 ]
       8  SAMPLE2        [SAMPLE PROGRAM 2 ]
       9  SAMPLE3        [SAMPLE PROGRAM 3 ]
      10  PROG001        [PROGRAM001       ]

  [ 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 will be displayed.

```
SAMPLE3
                                        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 [SHIFT] key.

Position data information

The coordinates and configuration for position data can be directly changed on the position data information screen.

```
| | | PAGE | CONF | DONE | [REPRE] | |
```

- F2 PAGE : Toggles between the standard axes and the extended axes.
- F3 CONFIG : Edits the configuration value.
- F4 DONE : Terminates changing the position data information.
- F5 [REPRE] : Toggles between Cartesian coordinates and joint coordinates.

Changing an instruction item

To change an instruction item, press the F4, [CHOICE] key to display the motion instruction item menu, then select an instruction item from the menu.

- Motion type: Controls a path to the end position (joint, linear, circular, circle arc). When the motion type is changed, the feed rate unit is also automatically changed.
- Position variable: The variable storing position data and the variable number are changed.
- Feed rate: The speed of the robot when it moves (robot motion speed) and the feed rate unit are changed.
- Positioning type: Positioning at the specified position is changed.
- Additional motion instruction: An additional instruction to be executed when the robot is moving is changed.

Procedure 5-18 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
                                           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 [SHIFT] key. The new position will be recorded.

```
SAMPLE1
                                        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  >
```

But when "Confirmation of TOUCHUP" is TRUE(refer to "3.15 SYSTEM CINFIG MENU"), you have to answer a question that system asks.

```
  Are you sure to TOUCHUP?
                          YES     NO
```

- YES : The position data will have been recorded.
- NO : The position data will have been recorded.

3 When the position data is taught to the motion instruction with an incremental option again, an incremental option will be removed.

```
SAMPLE1
                                        4/5

     4: L  P[3] 100mm/sec FINE INC
  [End]

   POINT                          TOUCHUP  >
```

```
  Delete Inc option and record position ?
                          YES     NO
```

- YES : An incremental option is removed and position data is taught.
- NO : The position data is not taught.

```
SAMPLE1
                                        4/5

     4: L  P[3] 100mm/sec 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
                                            5/6

     5: J  PR[5] 100% FINE
     [End]






     Position has been recorded to PR[5].

     | POINT |       |       |       | TOUCHUP | > |
```

Procedure 5-19 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 will be displayed.

```
SAMPLE1
      P[2] UF:0  UT:1          CONF:NDB 000
      X   1500.374   mm  W     40.000   deg
      Y   -342.992   mm  P     10.000   deg
      Z    956.895   mm  R     20.000   deg
    Position Detail

        2: J  P[2] 70% CNT50



     Enter value
     |       |       | CONF  | DONE  | [REPRE] |   |
```

2 To change the position, move the cursor to the coordinates for each axis and enter new coordinates.

```
SAMPLE1
      P[2] UF:0  UT:1          CONF:NDB 000
      X   1500.374   mm  W     40.000   deg
      Y   -300.000   mm  P     10.000   deg
      Z    956.895   mm  R     20.000   deg
    Position Detail

        2: J  P[2] 70% CNT50



     Enter value
     |       |       | CONF  | DONE  | [REPRE] |   |
```

3 To change the configuration value, press the F3, CONF key, move the cursor to the configuration field, then enter a new configuration value with the cursor keys (↑ and ↓).

```
SAMPLE1
      P[2] UF:0  UT:1          CONF:NDB 000
      X   1500.374   mm  W     40.000   deg
      Y   -300.000   mm  P     10.000   deg
      Z    956.895   mm  R     20.000   deg
    Position Detail

        2: J  P[2] 70% CNT50



     Select Flip or Non-flip by UP/DOWN key
                     POSITION   DONE   [REPRE]
```

4 To change a coordinate system, press the F5, [REPRE] key and select the coordinate system to be changed.

```
SAMPLE1
      P[2] UF:0  UT:1
      J1        0.125 deg  J4   -95.000 deg
      J2       23.590 deg  J5     0.789 deg
      J3       30.300 deg  J6  -120.005 deg
    Position Detail

        2: J  P[2] 70% CNT50



     Enter value
                                DONE   [REPRE]
```

> **NOTE**
> JOINT display is valid when the robot is adjusted to the zero-degree position or when non-kinematics operation such as table operation control is executed.

5 After changing position data information, press the F4, DONE key.

Procedure 5-20 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 menu of the instruction items, then select the instruction item to be changed from the menu.
The following screens show changing the motion type from linear motion to joint motion:

```
SAMPLE1

  Motion Modify 1   1000cm/min CNT30
  1 Joint
  2 Linear
  3 Circular
  4 Circle Arc
  5
  6
  7
  8

        5: L  P[5] 1000cm/min CNT30
                          [CHOICE]
```

```
SAMPLE1

      5: J  P[5] 100% CNT30
   [End]



   Enter value or press ENTER
 |  |  |  | [CHOICE] | POSITION |
```

3 The following screens show changing from the position variable to the position register.

```
SAMPLE1

 Motion Modify  1  100% CNT30
 1 P[ ]
 2 PR[ ]
 3
 4
 5
 6
 7
 8 --next page--

      5: J  P[5] 100% CNT30
 |  |  |  | [CHOICE] |  |
```

```
SAMPLE1

      5: J  PR[...] 100% CNT30
   [End]



   Enter value
 |  | DIRECT | INDIRECT | [CHOICE] | POSITION |
```

4 Change the feed rate.

```
SAMPLE1

      2: J  P[2] 70% FINE
   [End]



   Enter value
 | REGISTER |  |  | [CHOICE] |  |
```

5 Change the feed rate unit.

```
SAMPLE1

 Motion Modify 1   500cm/min CNT30
 1 mm/sec
 2 cm/min
 3 inch/min
 4 deg/sec
 5 sec
 6 msec
 7
 8 --next page--

      4: L  P[4] 500cm/min CNT30

                        [CHOICE]
```

```
SAMPLE1

      4: L  P[4] 500mm/min CNT30
    [End]

                        [CHOICE]
```

6 Change the positioning type.

```
SAMPLE1

 Motion Modify 1   70% FINE
 1 Fine
 2 Cnt
 3
 4
 5
 6
 7
 8

      4: J  P[4] 70% FINE

                        [CHOICE]
```

```
SAMPLE1

      4: J  P[4] 70% CNT100
    [End]

                        [CHOICE]
```

Procedure 5-21 Changing a circular motion instruction

Step

1 Place the cursor at the motion type of the circular motion instruction to be changed.
Press F4, [CHOICE]. The motion type list is displayed on the pop up menu.
The following screens show changing the circular motion instruction to the linear motion instruction.

```
SAMPLE1

        6: C  P[5]
              P[6] 500cm/min CNT30
     [End]

                          [CHOICE]
```

```
SAMPLE1

  Motion Modify 1
  1 Joint          500cm/min CNT30
  2 Linear
  3 Circular
  4 Circle Arc
  5
  6
  7
  8

        6: C  P[5]
                          [CHOICE]
```

```
SAMPLE1

        6: L  P[5] 500cm/min CNT30
        7: L  P[6] 500cm/min CNT30
     [End]

                        [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.
Press F4, [CHOICE] and select Circular.

```
SAMPLE1

 Motion Modify 1  500cm/min CNT30
 1 Joint
 2 Linear
 3 Circular
 4 Circle Arc
 5
 6
 7
 8

       6: L  P[6] 500cm/min CNT30

                         [CHOICE]
```

```
SAMPLE1

       6: C  P[6]
             P[...] 500cm/min CNT30
    [End]



    Enter value or press ENTER

                        [CHOICE] POSITION
```

NOTE
When a joint or linear motion instruction is changed to a circular motion instruction, the taught data for the end point of the arc is canceled.

Procedure 5-22 Adding and deleting an additional motion instruction

Step

1. Position the cursor to an additional motion instruction.
   Press F4, [CHOICE]. The additional motion instruction list will be displayed.
   To add an offset condition instruction, for example, follow the procedure below:

```
SAMPLE1

       7: L  P[2] 300mm/sec FINE
    [End]




                         [CHOICE]
```

```
SAMPLE1
                                   Motion Modify  3
       7: L        Motion Modify  2 AFTER
   Motion Modify  1   remental      NCE BEFORE
   1 No option        _Offset
   2 Wrist Joint      _Offset,PR[
   3 ACC              ependent EV
   4 Skip,LBL[]       ltaneous EV
   5 BREAK            E BEFORE
   6 Offset/Frames    ,LBL,PR       t page--
   7 Offset,PR[ ]     ext page--
   8 --next page--

       7: L  P[2] 300mm/sec FINE
                              [CHOICE]
```

```
SAMPLE1

       7: L  P[2] 300mm/sec FINE Offset
    [End]

                              [CHOICE]
```

2 To delete an offset instruction, for example, follow the procedure below:
Move the cursor to the Offset statement and press F4, [CHOICE]. The additional motion instruction list is displayed.
Select No Option. The Offset statement is deleted.

```
SAMPLE1

       7: L  P[2] 300mm/sec FINE Offset
    [End]

                              [CHOICE]
```

```
SAMPLE1
                                   Motion Modify  3
       7: L        Motion Modify  2 AFTER
   Motion Modify  1   remental      NCE BEFORE
   1 No option        _Offset
   2 Wrist Joint      _Offset,PR[
   3 ACC              ependent EV
   4 Skip,LBL[]       ltaneous EV
   5 BREAK            E BEFORE
   6 Offset/Frames    ,LBL,PR       t page--
   7 Offset,PR[ ]     ext page--
   8 --next page--

       7: L  P[2] 300mm/sec FINE Offset
                              [CHOICE]
```

```
SAMPLE1

     7: L  P[2] 300mm/sec FINE
   [End]

                              [CHOICE]
```

Procedure 5-23 Changing the move speed (between numeric specification and register specification)

```
SAMPLE1
                                                     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.

```
SAMPLE1
                                               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.)

```
SAMPLE1
                                               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,

```
SAMPLE1
                                            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.

```
SAMPLE1
                                            1/2
     1: J  P[1] ...% FINE
   [End]



   Enter value
 | REGISTER |  |  | [CHOICE] |  |
```

5 Enter a desired speed value (20 for example).

```
SAMPLE1
                                            1/2
     1: J  P[1] 20% FINE
   [End]




 |  |  |  | [CHOICE] |  |
```

5.4.3 Changing a Control Instruction

You can change the syntax, item, or variable of a control instruction.

Procedure 5-24 Changing a control instruction

Step

1 Move the cursor to the instruction item to be changed.

```
PROGRAM1
                                           11/20

    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.

```
PROGRAM1
                                         11/20
                  Wait statements 2
 Wait statements 1 ]
 1 R[ ]           ]
 2 On             ]
 3 Off            ]
 4 On+            ]
 5 Off-           ]
 6 DO[ ]
 7 DI[ ]          xt page--
 8 --next page--

     11:   WAIT RI[1]=ON

                          [CHOICE]
```

```
PROGRAM1
                                         11/20

     11:   WAIT RI[1]=R[...]
     12:   RO[1]=ON


    Enter value
           DIRECT   INDIRECT  [CHOICE]  [LIST]
```

```
PROGRAM1
                                         11/20

     11:   WAIT RI[1]=R[2]
     12:   RO[1]=ON



                          [CHOICE]
```

```
PROGRAM1
                                         11/20
 Wait statements 1 :[1]=R[2]
 1 <Forever>       N
 2 Timeout-LBL[ ]
 3 AND
 4 OR
 5
 6
 7
 8

     11:   WAIT RI[1]=R[2]

                          [CHOICE]
```

```
PROGRAM1
                                          11/20

     11:   WAIT RI[1]=R[2]
       :   TIMEOUT,LBL[...]
     12:   RO[1]=ON


   Enter value
        DIRECT  INDIRECT  [CHOICE]  [LIST]
```

```
PROGRAM1
                                          11/20

     11:   WAIT RI[1]=R[2] TIMEOUT,LBL[2]
     12:   RO[1]=ON

  [ INST ]                         [EDCMD]  >
```

5.4.4 Program Edit Instructions

The program edit instructions are used to edit a program.
Press the F5, [EDCMD] key to display the program edit instruction menu and select a desired program edit instruction from the menu.

```
                 EDCMD  2
                1 Icon Editor
  EDCMD  1      2
1 Insert        3
2 Delete        4
3 Copy          5
4 Find          6
5 Replace       7
6 Renumber      8
7 Comment       9
8 Undo          0 -- NEXT --
9 Remark
0 -- NEXT --
                    EDCMD   >
```

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 DO[1] is to be changed to DO[2] in the program.)

Renumber

Renumbers the position number in ascending order. Whenever a motion instruction is taught, the position number is increased regardless of its location in the program. When insertions and deletions are repeated, the position numbers are not sequentially arranged in a program. Renumbering arranges them sequentially in the program.

Comment

On the program editing screen, the user can choose whether to display or hide a comment for the instructions listed below. Note that no comment can be edited.

- DI instruction, DO instruction, RI instruction, RO instruction, GI instruction, GO instruction, AI instruction, AO instruction, UI instruction, UO instruction, SI instruction, SO instruction
- Register instructions
- Position register instructions (including position registers in the position data format for move instructions)
- Palletizing instructions
- Move instruction register speed specifications

The instructions listed below are always accompanied by a comment, and do not allow display switching but allow editing.

- Move instruction position variable
- Label instructions
- Force control instructions

> **NOTE**
> 1 The comment display area for an instruction item that is too long to be displayed on one line of the screen may be shortened.
> 2 No comment is displayed for a register indirect specification.
> PR[R [1]] = ...

Undo

Program edit operations such as an instruction modification, line insertion, and line deletion can be cancelled to return to the state present before those edit operations are performed. If an undo operation is performed during editing of a program line, all operations performed for that line are undone. For example, if a line is inserted or deleted, the state before the insertion or deletion operation is restored.
If an undo operation is immediately followed by another undo operation, the state present before the first undo operation is performed is restored.

> **NOTE**
> If an undo operation is performed for a line during program editing, all operations performed for that line are undone. This means that if an instruction is taught in a blank line or the last line of a program, and an undo operation is performed for that line during editing, the taught instruction is deleted.

Remark

Remarks or unremarks a program instruction. A remarked instruction is not executed at program execution. All program instructions can be remarked and unremarked.

- “//” is displayed in the head of line in which the remarked instruction is taught.
- The multiple instructions can be remarked and unremarked at the same time.
- Because the data in the remarked instruction is saved, the unremarked instruction can be executed immediately after unremark.
- The instruction which is copied from the remarked instruction is remarked.
- You can find and replace the remarked instruction as with normal instruction.
- You can renumber the remarked motion instruction by renumber operation in the program edit instruction.
- You can select whether the comment of I/O and other instruction is displayed or not by the Comment command in the program edit instruction.

NOTE

1 The remarked instruction can not be operated by the key. When the cursor is on the line number of the remarked instruction, even if the right arrow key is pressed, the cursor does not move to right.
2 The position data change by F5, TOUCHUP key cannot be executed in the remarked motion instruction.
3 If the label instruction is remarked, “*” is added to the label number. When the remarked label instruction is unremarked, “*” is deleted automatically. However, when the label instruction which has the same number exists in the program, “*” is not deleted.
4 When TEMP memory / TPP memory is insufficient, "TPIF-132 Can't recover this operation" "MEMO-126 No more available TPP memory" or "MEMO-151 No more available memory(TEMP)" may be caused and remarking may be failed. Delete unused TP program to increase the free memory space.

ICON Editor

Instead of the traditional program editor of robot controller using pop-up menus to edit instructions of a robot program, it is possible to edit the robot program by touching ICON on the screen. Please refer to “11.3.1 ICON Editor”.

NOTE

ICON editor can be used on only the teach pendant which equips the touch panel.

Color

It is possible to display the following instructions with color as Fig. 5.4.4 (a) on software version 7DC3 or later. You can switch Colored/Non colored by selecting this item.

- Application instruction: Displays Arc welding, Spot welding and so on instructions in RED. (line 9 on Fig. 5.4.4 (a))
- I/O instruction: Displays ON in Green, OFF in RED. (line 2, line 3 on Fig. 5.4.4 (a))
- Remark/Multi-lng Remark instruction: Displays Remarks in Yellow. (line 1, line 4 , line 7 on Fig. 5.4.4(a))

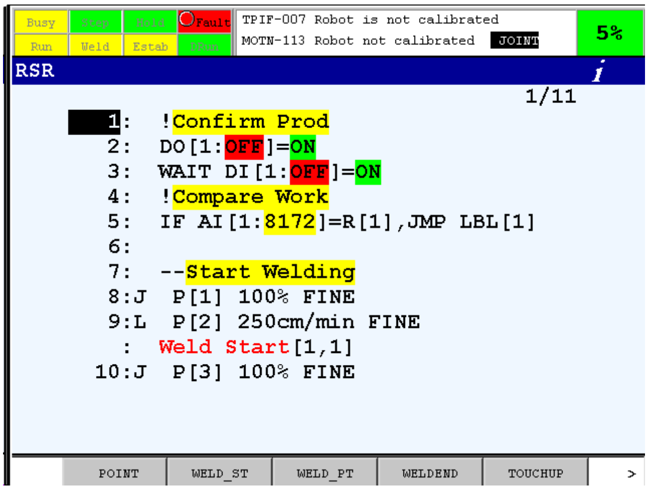


Fig. 5.4.4(a) Colored TP Edit Screen

IO status

It is possible to display I/O instructions with I/O's status as Fig. 5.4.4(a). You can switch Display/Non Display by selecting this item.

- Analog I/O, Group I/O: Displays the value of I/O in Yellow. (line 5 on Fig. 5.4.4 (a))
- The others: Displays ON (in Green)/OFF (in Red). (line 2, line 3 on Fig. 5.4.4 (a))

Procedure 5-25 Inserting blank lines

Step

1 Press the [NEXT] key, > to display F5, [EDCMD].

```
SAMPLE1
                                          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 will be displayed.

```
SAMPLE1
                                          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       EDCMD 1
       5: J  P[1] 100% FINE          1 Insert
    [End]                            2 Delete
                                     3 Copy
                                     4 Find
                                     5 Replace
                                     6 Renumber
                                     7 Comment
                                     8 Undo
                                     9 Remark
    [ INST ]                         [EDCMD]  >
```

3 Select Insert.

```
SAMPLE1
                                          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 lines 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.

```
SAMPLE1
                                          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
       7: J  P[1] 100% FINE
    [End]


    [ INST ]                         [EDCMD]
```

The two blank lines are inserted into the program and all the lines in the program are renumbered.

Procedure 5-26 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] key, > to display F5, [EDCMD].

```
SAMPLE1
                                           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.

```
SAMPLE1
                                           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          EDCMD 1
        5: J  P[1] 100% FINE              1 Insert
     [End]                                2 Delete
                                          3 Copy
                                          4 Find
                                          5 Replace
                                          6 Renumber
                                          7 Comment
                                          8 Undo
                                          9 Remark
  [ INST ]                        [EDCMD]  >
```

4 Select Delete.

```
SAMPLE1
                                            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 ↓).
6 To cancel deleting the selected line, press the F5, NO key. To delete the selected lines, press the F4, YES.

```
SAMPLE1
                                                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 5-27 Copying and pasting instructions

Step

1 Press the [NEXT] key, > until F5, [EDCMD].

```
SAMPLE1
                                                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 will be displayed.

```
SAMPLE1
                                          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      EDCMD 1
      5: J  P[1] 100% FINE           1 Insert
   [End]                             2 Delete
                                     3 Copy
                                     4 Find
                                     5 Replace
                                     6 Renumber
                                     7 Comment
                                     8 Undo
                                     9 Remark
  [ INST ]                        [EDCMD]  >
```

3 Select 3 Copy.
The following screens show copying 2nd to 4th lines to 5th to 7th lines.

```
SAMPLE1
                                         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.
As a result of above steps, the selected instructions (2nd to 4th lines in this example) were copied in memory.

```
SAMPLE1
      1: J  P[1] 10
      2: J  P[2] 70
      3: L  P[3] 10
      4: L  P[4] 50
      5: J  P[1] 10
   [End]

   Move cursor to s
            COPY
```

```
SAMPLE1
      1: J  P[1] 10
      2: J  P[2] 70
      3: L  P[3] 10
      4: L  P[4] 50
      5: J  P[1] 10
   [End]

   Move cursor to s
            COPY
```

```
SAMPLE1
                                         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[1] 100% FINE
   [End]

   Select lines
            COPY                  PASTE
```

5 Decide where you want to paste the sentences copied in the memory.

```
SAMPLE1
                                          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[1] 100% FINE
   [End]

   Paste before this line ?
|  |        | LOGIC | POSID | POSITION | CANCEL | > |
```

```
   Paste reversed order before this line ?
|  | R-LOGIC | R-POSID | RM-POSID | R-POS | RM-POS | > |
```

6 Select the copying and pasting method (copying from the original).
In this example, F3, POSID is selected.

```
SAMPLE1
                                          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 |  |
```

The instructions copied in the memory are inserted.

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.

Pasting methods

The following copying and pasting methods are provided:

```
   Paste before this line ?
|  |        | LOGIC | POSID | 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:

```
   Paste reversed order before this line ?
|  | R-LOGIC | R-POSID | RM-POSID | R-POS | RM-POS | > |
```

The selected instructions are copied in reverse order.

F3 and F5 have the following functions:

- F3, RM-POS-ID: Copies the move instructions at a copy source in reverse order without changing the position numbers of the move instructions. The move type, move speed, and so forth of each move instruction are changed so that a movement totally opposite to the movement of the copy source is made.
- F5, RM-POS : Copies the move instructions at a copy source in reverse order. Then assigns new position numbers. The move type, move speed, and so forth of each move instruction are changed so that a movement totally opposite to the movement of the copy source is made.

NOTE
The copy function for a reverse movement is not supported for the additional move instructions listed below. If the move instructions at a copy source include any of the move instructions below, RM-POS-ID or RM-POS generates a warning, and only a copy operation in reverse order is performed.

- Application instruction
- Skip and high-speed skip instructions
- Incremental instruction
- Continuous rotation instruction
- Pre-execution/post-execution instruction
- Multi-group operation

Example When the F4, R-POS is pressed

```
SAMPLE1
                                          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: L  P[7] 500mm/sec FINE
     6: L  P[6] 1000cm/min CNT30
     7: J  P[5] 70% CNT50
     8: J  P[1] 100% FINE
   [End]

   Select lines
        COPY                PASTE
```

Procedure 5-28 Cutting and pasting instructions

Step

1 Press the [NEXT] key, > until F5, [EDCMD].

```
SAMPLE1
                                        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 will be displayed.

```
SAMPLE1
                                        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          EDCMD 1
      5: J  P[1] 100% FINE             1 Insert
   [End]                               2 Delete
                                       3 Copy/Cut
                                       4 Find
                                       5 Replace
                                       6 Renumber
                                       7 Comment
                                       8 Undo
                                       9 Remark

    [ INST ]                        [EDCMD]  >
```

3 Select 3 Copy/Cut.
The following screens show copying 2nd to 3rd lines to 5th to 6th lines.

```
SAMPLE1
                                       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

              SELECT              PASTE
```

4 Select the start line to be cut, and select F2, [SELECT] key.

```
SAMPLE1
                                        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]

  Move cursor to select range
|   |         | COPY | CUT |        | PASTE |   |
```

5 Select the range, and select F3, [CUT] key.

```
SAMPLE1

     1: J  P[1] 100% FINE
     2: J  P[2] 70% CNT50
     3: L P[3] 1000cm/min CNT
     4: L P[4] 500mm/sec FINE
     5: J  P[1] 100% FINE
  [End]

  Move cursor to select range
|   |       | COPY | CUT |       |
```

```
SAMPLE1
                                        2/4
     1: J  P[1] 100% FINE
     2: L  P[4] 500mm/sec FINE
     3: J  P[1] 100% FINE
  [End]

  Select lines
|   |       | SELECT |      |      | PASTE |   |
```

As a result of above steps, the selected instructions (2nd to 3rd lines in this example) were cut and copied in memory.

6 Decide where to paste the sentences copied in the memory.

```
SAMPLE1
                                        3/4
     1: J  P[1] 100% FINE
     2: L  P[4] 500mm/sec FINE
     3: J  P[1] 100% FINE
  [End]

  Paste before this line ?
|   |         | LOGIC   | POSID    | POSITION | CANCEL | > |
|   | R-LOGIC | R-POSID | RM-POSID | R-POS    | RM-POS | > |
```

7 Select the pasting method (copying from the original).
In this example, F3, POSID is selected.

```
SAMPLE1
                                         5/6
      1: J  P[1] 100% FINE
      2: L  P[4] 500mm/sec FINE
      3: J  P[2] 70% CNT50
      4: L  P[3] 1000cm/min CNT30
      5: J  P[1] 100% FINE
   [End]


   Select lines
|    | SELECT |        |        | PASTE |   |
```

The instructions copied in the memory are inserted.

8 By repeating the above steps 6 to 7, the same instruction group can be pasted at any number of locations in the program.

9 To terminate the pasting of instructions, press the [PREV] key.

Please refer to the “Pasting methods” in “Procedure 5-27 Copying and pasting instructions” about the kinds of pasting methods.

Procedure 5-29 Finding a program instruction item

Step

1 Press the [NEXT] key, > until F5, [EDCMD].

```
SAMPLE3
                                         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 will be displayed.

```
SAMPLE3
                                         1/10
      1: J  P[1] 100% FINE
      2:   R[1]=0
      3:   LBL[1]                          EDCMD 1
      4: L  P[2] 1000cm/min CNT30        1 Insert
      5: L  P[3] 500mm/sec FINE          2 Delete
      6:   IF DI[1]=ON JMP LBL[2]        3 Copy
      7:   R[1]=R[1]+1                   4 Find
      8:   JMP LBL[1]                    5 Replace
      9:   LBL[2]                        6 Renumber
   [End]                                 7 Comment
                                         8 Undo
                                         9 Remark
| [ INST ] |        |        |        | [EDCMD] | > |
```

3 Select Find.

4 Select a program instruction item to be found. The following screens show how to find instruction, LBL[1].

```
Select Find menu 1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 Miscellaneous
7 CALL
8 --next page--
```

```
Select Find menu 2
1 Program control
2 Skip
3 Offset/Frames
4 Tool_Offset
5 Multiple control
6 MONITOR/MON. END
7 DIAGNOSE
8 --next page--
```

```
JMP statement    1
1 JMP LBL[  ]
2 LBL[  ]
3
4
5
6
7
8
```

```
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.

```
SAMPLE3
                                         3/10
        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]

     Find item
                              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.

```
SAMPLE3
                                         9/10
        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]

     Find item
                              NEXT    EXIT
```

7 To terminate finding an instruction, press the F5, EXIT key.

> **NOTE**
> The position of a track/offset instruction or touch sensor instruction cannot be found using the search instruction.

Procedure 5-30 Replacing a program instruction item

Step

1 Press the [NEXT] key, > until F5, [EDCMD].

```
SAMPLE3
                                              1/9
        1: J  P[1] 100% FINE
        2: J  P[2] 70% CNT50
        3:   LBL[1]
        4: L  P[2] 1000cm/min CNT30
        5: L  P[3] 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 editing instruction menu will be displayed.

```
SAMPLE3
                                              1/9
        1: J  P[1] 100% FINE
        2: J  P[2] 70% CNT50
        3:   LBL[1]                        EDCMD 1
        4: L  P[2] 1000cm/min CNT30      1 Insert
        5: L  P[3] 500mm/sec FINE        2 Delete
         :   Skip,LBL[2]                 3 Copy
        6:   JMP LBL[1]                  4 Find
        7:   LBL[2]                      5 Replace
        8: J  P[5] 100% FINE             6 Renumber
     [End]                               7 Comment
                                         8 Undo
                                         9 Remark

   [ INST ]                            [EDCMD]  >
```

3 Select Replace.

4 Select a program instruction item to be replaced and press the [ENTER] key.
On the screen below the feed rate specified in the motion instruction is changed to another value.

```
Select Replac menu 1
1 Registers
2 Motion modify
3 I/O
4 JMP/LBL
5 CALL
6 TIME BEFORE/AFTE
7
8
```

```
Modify motion menu 1
1 Replace speed
2 Replace term
3 Insert option
4 Remove option
5
6
7
8
```

The following replacement items are displayed:

- Replace speed: Changes the feed rate to another value.
- Replace term: Changes the positioning type to another type
- Insert option: Inserts an additional motion instruction.
- Remove option: Deletes an additional motion instruction.

5 Select Replace speed.

```
Select interpolate 1
1 Unspecified type
2 J
3 L
4 C
5 A
6
7
8
```

- Unspecified type: Changes the feed rates in all motion instructions.
- J: Changes the feed rates only in motion instructions for joint control.
- L: Changes the feed rates only in motion instructions for linear control.
- C: Changes the feed rates only in motion instructions for circular control.
- A: Changes the feed rates only in motion instructions for circle arc control.

6 Specify the target type of the operation instruction.

```
Speed type menu   1
1 All type
2 Speed value
3 R[ ]
4 R[R[ ]]
5
6
7
8
```

- 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.

```
Select motion item 1
1 %
2 mm/sec
3 cm/min
4 inch/min
5 deg
6 sec
7 msec
8
```

8 Specify a target speed unit.

```
Speed type menu   1
1 Speed value
2 R[ ]
3 R[R[ ]]
4
5
6
7
8
```

- 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 feed rate is to be changed.
The following screen shows the procedure when the item 1 "Speed value" is selected.

```
     Enter speed value:
```

10 Enter a desire feed rate.

```
SAMPLE3
                                              1/9
        1: J  P[1] 100% FINE
        2: J  P[2] 70% CNT50
        3:   LBL[1]
        4: L  P[2] 1000cm/min CNT30
        5: L  P[3] 500mm/sec FINE
         :   Skip,LBL[2]
        6:   JMP LBL[1]
        7:   LBL[2]
        8: J  P[5] 100% FINE
     [End]

     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.

11 Select a replacement method. The following screen shows the result when ALL is selected.

```
SAMPLE3
                                        1/9
     1: J  P[1] 50% FINE
     2: J  P[2] 50% CNT50
     3:   LBL[1]
     4: L  P[2] 1000cm/min CNT30
     5: L  P[3] 500mm/sec FINE
      :   Skip,LBL[2]
     6:   JMP LBL[1]
     7:   LBL[2]
     8: J  P[5] 50% FINE
  [End]

   [ INST ]                    [EDCMD]  >
```

12 To terminate item replacement, press the F5, EXIT.

> ⚠ **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 5-31 Renumbering the position number

Step

1 Press the [NEXT] key, >, then press the F5, [EDCMD].

```
SAMPLE1
                                       1/8
     1: J  P[8] 100% FINE
     2: J  P[6] 70% CNT50
     3: L  P[3] 1000cm/min CNT30
     4: L  P[5] 500mm/sec FINE
     5: J  P[1] 100% FINE
     6: L  P[5] 500mm/sec FINE
     7: J  P[8] 100% FINE
  [End]

   [ INST ]                    [EDCMD]  >
```

2 Press F5, [EDCMD]. The editing instruction menu will be displayed.

```
SAMPLE1
                                         1/8
       1: J  P[8] 100% FINE
       2: J  P[6] 70% CNT50
       3: L  P[3] 1000cm/min CNT30   EDCMD 1
       4: L  P[5] 500mm/sec FINE     1 Insert
       5: J  P[1] 100% FINE          2 Delete
       6: L  P[5] 500mm/sec FINE     3 Copy
       7: J  P[8] 100% FINE          4 Find
    [End]                            5 Replace
                                     6 Renumber
                                     7 Comment
                                     8 Undo
                                     9 Remark
     [ INST ]                    [EDCMD]  >
```

3 Select Renumber.

```
    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.

```
SAMPLE1
                                         1/8
       1: J  P[1] 100% FINE
       2: J  P[2] 70% CNT50
       3: L  P[3] 1000cm/min CNT30
       4: L  P[4] 500mm/sec FINE
       5: J  P[5] 100% FINE
       6: L  P[4] 500mm/sec FINE
       7: J  P[1] 100% FINE
    [End]

     [ INST ]                    [EDCMD]  >
```

Procedure 5-32 Comment display switching

Step

1 Press the [NEXT] key to display F5, [EDCMD].

```
PNS0001
                                         1/10
       1:   R[1]=DI[2]
       2:   DO[3]=ON
       3:   R[R[1]]=DI[R[2]]
       4:   PR[1]=P[3]
       5:   PR[1,2]=PR[R[3],R[4]]
       6:   PL[1]=PL[R[3]]
       7: J  PR[1] 100% FINE
       8: J  P[1] 100% FINE
       9:   LBL[1]
    [End]

     [ INST ]                    [EDCMD]  >
```

2 Press F5, [EDCMD] to display the editing instruction menu.

```
PNS0001
                                              1/10
      1:   R[1]=DI[2]
      2:   DO[3]=ON
      3:   R[R[1]]=DI[R[2]]             EDCMD 1
      4:   PR[1]=P[3]                 1 Insert
      5:   PR[1,2]=PR[R[3],R[4]]      2 Delete
      6:   PL[1]=PL[R[3]]             3 Copy
      7: J  PR[1] 100% FINE           4 Find
      8: J  P[1] 100% FINE            5 Replace
      9:   LBL[1]                     6 Renumber
   [End]                              7 Comment
                                      8 Undo
                                      9 Remark

   [ INST ]                           [EDCMD]  >
```

3 Select Item 7 Comment.

```
PNS0001
                                              1/10
      1:   R[1:Comment]=DI[2:Comment]
      2:   DO[3:Comment]=ON
      3:   R[R[1]]=DI[R[2]]
      4:   PR[1:Comment]=P[3:Comment]
      5:   PR[1,2:Comment]=PR[R[3],R[4]]
      6:   PL[1:Comment]=PL[R[3]]
      7: J  PR[1:Comment] 100% FINE
      8: J  P[1:Comment] 100% FINE
      9:   LBL[1:Comment]
   [End]

   [ INST ]                           [EDCMD]  >
```

4 To disable comment display, select Comment of the function key F5, [EDCMD] again.

Procedure 5-33 Undoing edit operations

Step

1 Press the [NEXT] key to display F5, [EDCMD].

```
PNS0001
                                              1/10
      1:   R[1]=DI[2]
      2:   DO[3]=ON
      3:   R[R[1]]=DI[R[2]]
      4:   PR[1]=P[3]
      5:   PR[1,2]=PR[R[3],R[4]]
      6:   PL[1]=PL[R[3]]
      7: J  PR[1] 100% FINE
      8: J  P[1] 100% FINE
      9:   LBL[1]
   [End]

   [ INST ]                           [EDCMD]  >
```

2 Press F5, [EDCMD] to display the editing instruction menu.

```
PNS0001
                                          1/10
      1:   R[1]=DI[2]
      2:   DO[3]=ON
      3:   R[R[1]]=DI[R[2]]
      4:   PR[1]=P[3]                        EDCMD  1
      5:   PR[1,2]=PR[R[3],R[4]]          1 Insert
      6:   PL[1]=PL[R[3]]                 2 Delete
      7: J  PR[1] 100% FINE               3 Copy
      8: J  P[1] 100% FINE                4 Find
      9:   LBL[1]                         5 Replace
   [End]                                  6 Renumber
                                          7 Comment
                                          8 Undo
                                          9 Remark

   [ INST ]                            EDCMD   >
```

3 Select Item 8 Undo.

```
PNS0001
                                          1/10
      1:   R[1]=DI[2]
      2:   DO[3]=ON
      3:   R[R[1]]=DI[R[2]]
      4:   PR[1]=R[3]
      5:   PR[1,2]=PR[R[3],R[4]]
      6:   PL[1]=PL[R[3]]
      7: J  PR[1] 100% FINE
      8: J  P[1] 100% FINE
      9:   LBL[1]
   [End]

   Undo?(Edit)

                                 YES     NO
```

4 To perform an undo operation, select F4, YES. To cancel the undo operation, select F5, NO. When F4, YES is selected, the edit operation is undone.

```
PNS0001
                                          1/10
      1:   R[1]=RI[4]
      2:   DO[3]=ON
      3:   R[R[1]]=DI[R[2]]
      4:   PR[1]=P[3]
      5:   PR[1,2]=PR[R[3],R[4]]
      6:   PL[1]=PL[R[3]]
      7: J  PR[1] 100% FINE
      8: J  P[1] 100% FINE
      9:   LBL[1]
   [End]

   [ INST ]                            [EDCMD]  >
```

5 When an additional undo operation is performed in succession, the first undo operation performed can be cancelled; this means the state present before the first undo operation is performed is restored.

NOTE
If an edit operation is performed after an undo operation, the undo operation cannot be cancelled.

> ⚠ **CAUTION**
> An undo operation automatically rewrites the program, so that the results may not be those expected by the operator. Before executing a program after an undo operation, carefully check the program.

- This function can undo the following operations:
  a) Instruction modifications
  b) Line insertion
  c) Line deletion
  d) Copying of program statements (reading)
  e) Copying of program statements (insertion)
  f) Program instruction replacement
  g) Reassignment of position numbers
- An undo operation cancels all edit operations performed on the line where the cursor is currently placed, and restores the state 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

Procedure 5-34 Remarking

Step

1 Press the [NEXT] key to display F5, [EDCMD].

```
PNS0001
                                        1/10
      1:   R[1]=DI[2]
      2:   DO[3]=ON
      3:   R[R[1]]=DI[R[2]]
      4:   PR[1]=P[3]
      5:   PR[1,2]=PR[R[3],R[4]]
      6:   PL[1]=PL[R[3]]
      7: J  PR[1] 100% FINE
      8: J  P[1] 100% FINE
      9:   LBL[1]
   [End]

 [ INST ]                    [EDCMD]  >
```

2 Press F5, [EDCMD] to display the editing instruction menu.

```
PNS0001
                                        1/10
      1:   R[1]=DI[2]
      2:   DO[3]=ON
      3:   R[R[1]]=DI[R[2]]
      4:   PR[1]=P[3]                  EDCMD  1
      5:   PR[1,2]=PR[R[3],R[4]]     1 Insert
      6:   PL[1]=PL[R[3]]            2 Delete
      7: J  PR[1] 100% FINE          3 Copy
      8: J  P[1] 100% FINE           4 Find
      9:   LBL[1]                    5 Replace
   [End]                             6 Renumber
                                     7 Comment
                                     8 Undo
                                     9 Remark

 [ INST ]                     EDCMD   >
```

3 Select Item 9 Remark.

```
PNS0001
                                        1/10
      1:   R[1]=DI[2]
      2:   DO[3]=ON
      3:   R[R[1]]=DI[R[2]]
      4:   PR[1]=P[3]
      5:   PR[1,2]=PR[R[3],R[4]]
      6:   PL[1]=PL[R[3]]
      7: J  PR[1] 100% FINE
      8: J  P[1] 100% FINE
      9:   LBL[1]
   [End]

   Select lines to remark or unremark

                         REMARK  UNREMARK
```

4 Select lines to remark by the cursor key.

```
PNS0001
                                     1/10
      1:   R[1]=DI[2]
      2:   DO[3]=ON
      3:   R[R[1]]=DI[R[2]]
      4:   PR[1]=P[3]
      5:   PR[1,2]=PR[R[3],R[4]]
      6:   PL[1]=PL[R[3]]
      7: J  PR[1] 100% FINE
      8: J  P[1] 100% FINE
      9:   LBL[1]
   [End]

   Select lines to remark or unremark

                    REMARK   UNREMARK
```

5 Press F4, REMARK. The selected lines will be remarked.

```
PNS0001
                                     3/10
      1:   //R[1]=DI[2]
      2:   //DO[3]=ON
      3:   R[R[1]]=DI[R[2]]
      4:   PR[1]=P[3]
      5:   PR[1,2]=PR[R[3],R[4]]
      6:   PL[1]=PL[R[3]]
      7: J  PR[1] 100% FINE
      8: J  P[1] 100% FINE
      9:   LBL[1]
   [End]

 [ INST ]                     [EDCMD]  >
```

To unremark the remarked lines, select F5, UNREMARK.

NOTE
When you remark lines, any lines already remarked are skipped. When you unremark lines, any lines not remarked are skipped.

5.5 PROGRAM OPERATION

This section describes the following program operations:

- Changing program information
- Deleting a program
- Copying a program
- Displaying the attribute of a program

5.5.1 Changing Program Information

The program header information is changed with a program detail screen (see Section 4.1). Setting without the motion group can be done. The following items can be set:

- Program name: Name of program to be changed.
- Subtype: The subtype of a program to be changed.
- Comments: The comments in the program to be changed.
- Group mask: Specifies a motion group to be controlled in a program. You can also set a program having no motion group.

- Write protection: Prevents the modification of a program.
- Interruption disable: Causes a program that has no motion group not to be paused by the alarm whose severity is SERVO or lower, the emergency stop, and the hold.
- Stack size: Specifies a memory size used by the sub program call execution.

Display the following items on the program information screen:

- Creation Date
- Modification Date
- Name of the file to be copied
- Positions: FALSE/TRUE
- Memory area size of program

Deleting a program

The unnecessary program can be deleted.

Copying a program

The program with another name in the same content can be reproduced.

Display of a program attribute

The following program header information can be displayed on the program 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.
- Name Only - Only the name of program 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 5-35 Changing program information

Condition

- ■ The teach pendant must be enabled.

Step

1. Press [MENU] key to display the screen menu.
2. Select SELECT. The program selection screen will be displayed.
   Alternatively, press the SELECT key to display the program selection screen.

```
Select
                 61092 bytes free      1/11
      No. Program name         Comment
       1  -BCKEDT-         [                 ]
       2  GETDATA      MR [Get PC Data      ]
       3  REQMENU      MR [Request PC Menu  ]
       4  SENDDATA     MR [Send PC Data     ]
       5  SENDEVNT     MR [Send PC Event    ]
       6  SENDSYSV     MR [Send PC Sysvar   ]
       7  SAMPLE1         [SAMPLE PROGRAM 1 ]
       8  SAMPLE2         [SAMPLE PROGRAM 2 ]
       9  SAMPLE3         [SAMPLE PROGRAM 3 ]
      10  PROG001         [PROGRAM001       ]

 [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR ]  >
```

3 Press the [NEXT] key, > to display the next page, then press the F2, DETAIL key. The program information screen will be displayed.

```
Program detail
                                        1/7
      Creation Date:         16-Jan-1994
      Modification Date:     28-Mar-1994
      Copy Source:

      Positions: FALSE  Size:      312 Byte

      Program name:
       1 SAMPLE3
       2  Sub Type:      [None             ]
       3  Comment:       [SAMPLE PROGRAM 3 ]
       4  Group Mask:    [1,*,*,*,*,*,*,*  ]
       5  Write protect: [OFF              ]
       6  Ignore pause:  [OFF              ]
       7  Stack size:    [     500         ]

     END     PREV    NEXT
```

4 Specify each item (see Section 4.1).
If the motion instruction is taught in the program, you cannot set the 4 "Group Mask:" of this program.
5 After specifying program information, press the F1, END key.

Procedure 5-36 Deleting a program

Step

1 Press [MENU] key to display the screen menu.
2 Select SELECT. The program selection screen will be displayed.
The program selection screen can also be displayed by pressing the SELECT key, instead of executing steps 1 and 2 above.

```
Select
                    61092 bytes free      9/11
      No. Program name         Comment
       1  -BCKEDT-       [                 ]
       2  GETDATA     MR [Get PC Data      ]
       3  REQMENU     MR [Request PC Menu  ]
       4  SENDDATA    MR [Send PC Data     ]
       5  SENDEVNT    MR [Send PC Event    ]
       6  SENDSYSV    MR [Send PC Sysvar   ]
       7  SAMPLE1        [SAMPLE PROGRAM 1 ]
       8  SAMPLE2        [SAMPLE PROGRAM 2 ]
       9  SAMPLE3        [SAMPLE PROGRAM 3 ]
      10  PROG001        [PROGRAM001       ]

   [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR ]  >
```

3 Move the cursor to the name of a program to be deleted, then press the F3 DELETE key.

```
Select

       9  SAMPLE3        [SAMPLE PROGRAM 3 ]

     Delete OK ?

                                 YES     NO
```

4 Press the F4, YES key.
5 The specified program is deleted.

```
Select
                  61092 bytes free      9/10
      No. Program name         Comment
       1  -BCKEDT-       [                  ]
       2  GETDATA     MR [Get PC Data       ]
       3  REQMENU     MR [Request PC Menu   ]
       4  SENDDATA    MR [Send PC Data      ]
       5  SENDEVNT    MR [Send PC Event     ]
       6  SENDSYSV    MR [Send PC Sysvar    ]
       7  SAMPLE1        [SAMPLE PROGRAM 1  ]
       8  SAMPLE2        [SAMPLE PROGRAM 2  ]
       9  PROG001        [PROGRAM001        ]
      10  PROG002        [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 5-37 Copying a program

Step

1 Press [MENU] key to display the screen menu.
2 Select SELECT. The program selection screen will be displayed.
3 Press F1, COPY on the next page and then a program copy screen is displayed.

```
PROGRAM COPY
       ---   Copy Teach Pendant Program   1/1
      FROM:
        SMAPLE3
      To:
        SAMPLE3

                                          Alpha input 1
                                            Words
                                            Upper Case
                                            Lower Case
      Old Value:  SAMPLE3                   Options

   ABCDEF  GHIJKL  MNOPQR  STUVWX  YZ_@*.
```

4 Enter the name of the program to be copied, then press the [ENTER] key.

```
PROGRAM COPY
       ---   Copy Teach Pendant Program   1/1
      FROM:
        SMAPLE3
      To:
        PROGRAM1

      Copy OK ?

                           YES     NO
```

5 Press the F4, YES key.
6 The desired program is copied to the specified program, PROGRAM1.

```
Select
                 61092 bytes free     10/10
      No. Program name         Comment
       1  -BCKEDT-        [                  ]
       2  GETDATA      MR [Get PC Data       ]
       3  REQMENU      MR [Request PC Menu   ]
       4  SENDDATA     MR [Send PC Data      ]
       5  SENDEVNT     MR [Send PC Event     ]
       6  SENDSYSV     MR [Send PC Sysvar    ]
       7  SAMPLE1         [SAMPLE PROGRAM 1  ]
       8  SAMPLE2         [SAMPLE PROGRAM 2  ]
       9  SAMPLE3         [SAMPLE PROGRAM 3  ]
      10  PROGRAM1        [SAMPLE PROGRAM 3  ]

   [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR ]  >
```

Procedure 5-38 Displaying the Attribute of the Program

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select SELECT. The program selection screen will be displayed.
You can select a program selection screen by pressing [SELECT] key instead of the above 1 to 2 procedure.

```
Select
                 61092 bytes free      7/10
      No. Program name         Comment
       1  -BCKEDT-        [                  ]
       2  GETDATA      MR [Get PC Data       ]
       3  REQMENU      MR [Request PC Menu   ]
       4  SENDDATA     MR [Send PC Data      ]
       5  SENDEVNT     MR [Send PC Event     ]
       6  SENDSYSV     MR [Send PC Sysvar    ]
       7  SAMPLE1         [SAMPLE PROGRAM 1  ]
       8  SAMPLE2         [SAMPLE PROGRAM 2  ]
       9  PROG001         [PROGRAM001        ]
      10  PROG002         [PROGRAM002        ]

   [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR ]  >
```

3 Press F5, [ATTR].

```
Select
                 61092 bytes free      7/10
      No. Program name         Comment
       1  -BCKEDT-        [                  ]
       2  GETDATA      MR [Get PC Data       ]
       3  REQMENU      MR [Request PC Menu   ]
       4  SENDDATA     MR [Send PC        ATTR      1
       5  SENDEVNT     MR [Send PC  1 Comment
       6  SENDSYSV     MR [Send PC  2 Protection
       7  SAMPLE1         [SAMPLE   3 Last Modified
       8  SAMPLE2         [SAMPLE   4 Size
       9  PROG001         [PROGRAM  5 Copy Source
      10  PROG002         [PROGRAM  6 Name Only

   [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR ]  >
```

4 Select Size.

5 The number of lines and size of a program will be displayed at the place that the comments are displayed.

```
Select
                 61092 bytes free      1/10
      No. Program name         Comment
       1  -BCKEDT-       [      0/       132]
       2  GETDATA     MR [      2/       286]
       3  REQMENU     MR [      2/       262]
       4  SENDDATA    MR [      2/       288]
       5  SENDEVNT    MR [      2/       224]
       6  SENDSYSV    MR [      2/       274]
       7  SAMPLE1        [     32/       839]
       8  SAMPLE2        [     12/      1298]
       9  PROG001        [     20/       786]
      10  PROG002        [      8/       449]

  [ 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

```
a  ------------------> AAA  LINE 0  T1 RUNNING  JOINT     10%
b  --> BBB
c  ------------> <<BACKGROUND>>                      1/3
            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.

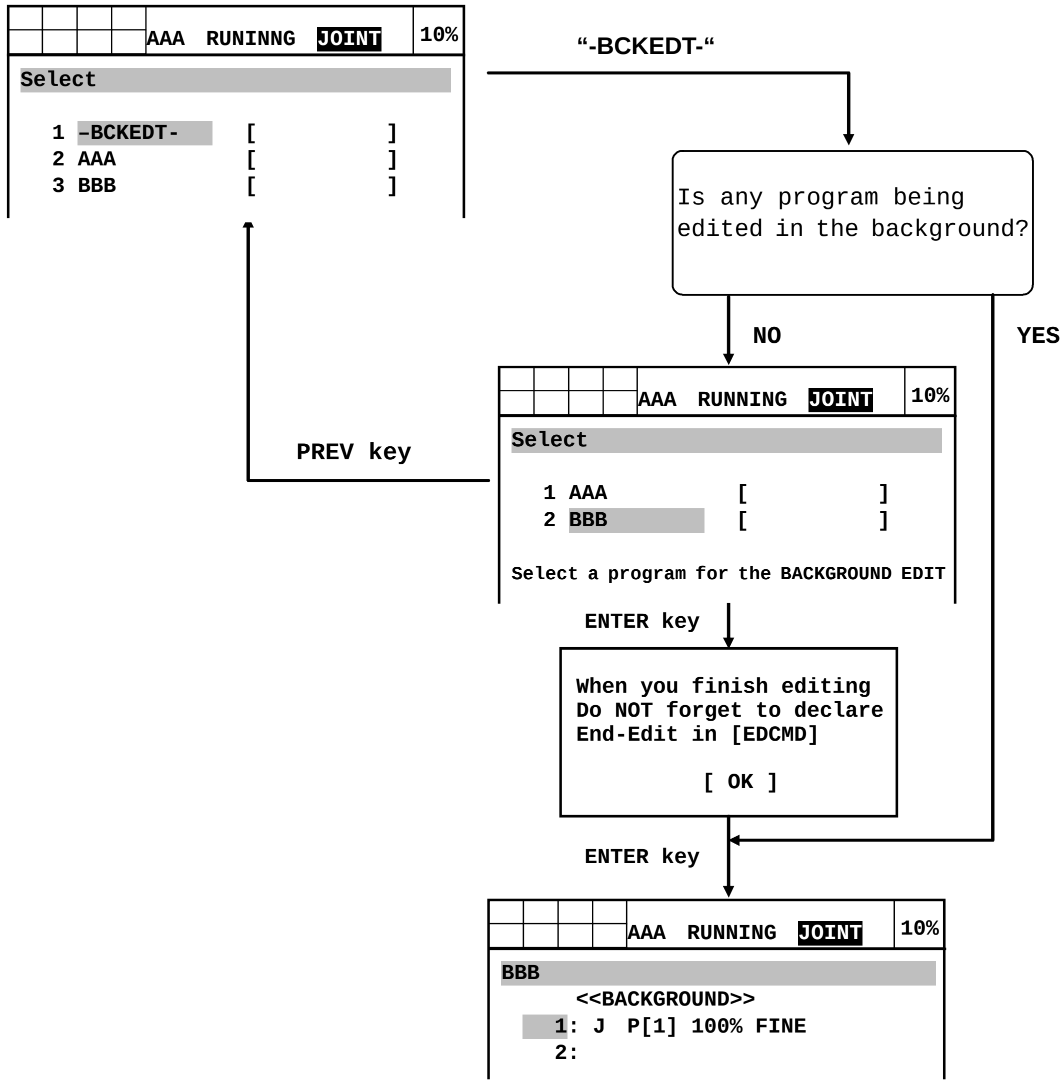
- To terminate background editing, press the F5, [EDCMD] key on the edit screen to display a menu, then select End_edit from the displayed menu.
  Here, the user can choose whether to reflect the results of background editing in the original program or discard the results of background editing.
- No 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

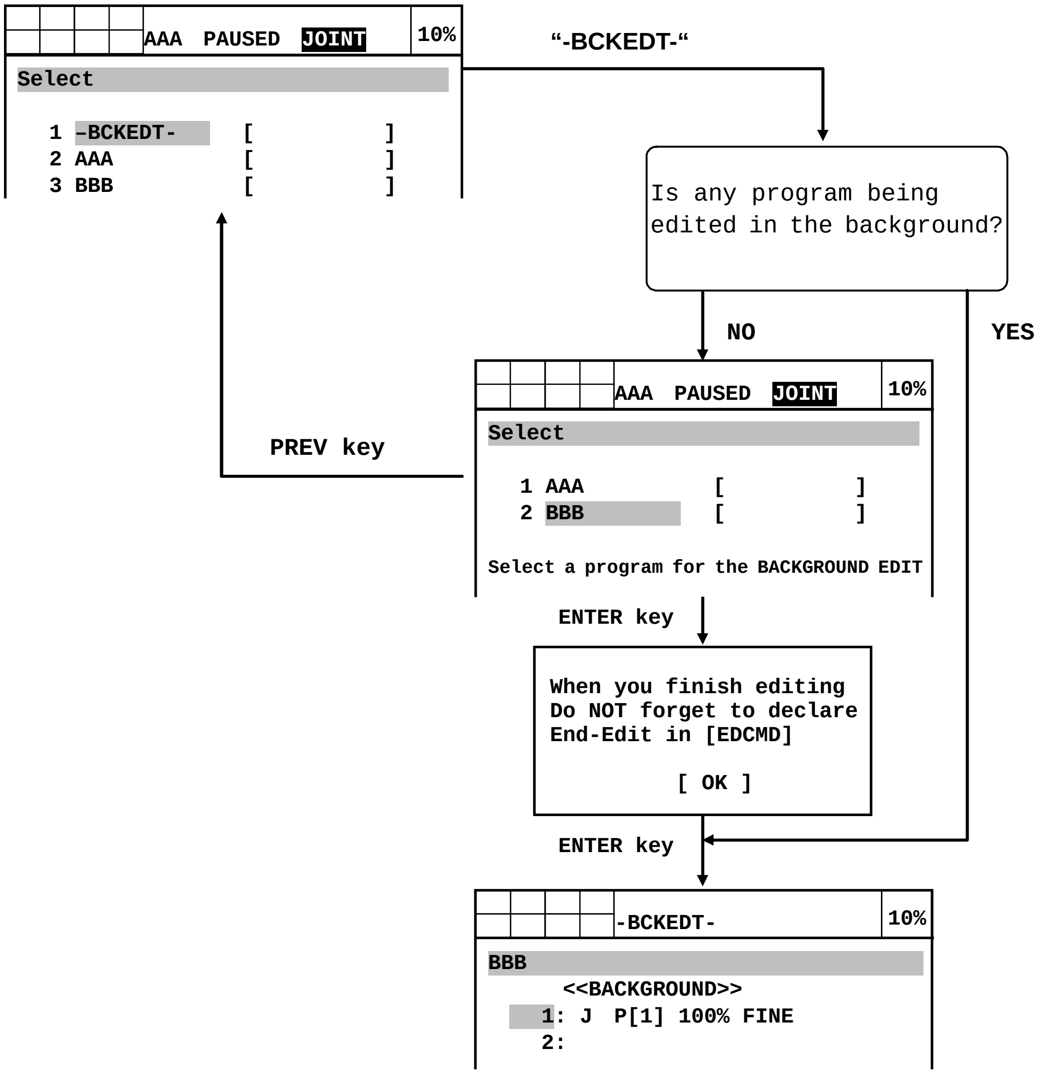
When background editing is started with the teach pendant disabled

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

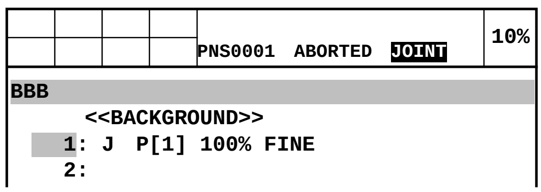
If the special program for background editing is selected when the teach pendant is enabled, the program is selected in the foreground, and its test execution is enabled.



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  JOINT    10%
BBB
      <<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.

Teach pendant : Disable

```
PNS0001  RUNNING  JOINT    10%
BBB
      <<BACKGROUND>>
    1: J  P[1] 100% FINE
    2:
```

→

Teach Pendant : Enable

```
PNS0001  PAUSED   JOINT    10%
PNS0001
      <<BACKGROUND>>
    1:
    2:
```

Disable the teach pendant, then
a. Press the EDIT key on the program edit screen.
b. Select “-BCKEDT-“ on the program list screen.

If no program is selected in the foreground, the special program (”-BCKEDT-”) is selected to allow the program being edited in the background to be executed. The status line displays the state of ”-BCKEDT-”.

Teach pendant : Disable

```
                  JOINT    10%
BBB
      <<BACKGROUND>>
    1:
    2:
```

→

Teach pendant : Enable

```
-BCKEDT-        JOINT    10%
BBB
      <<BACKGROUND>>
    1:
    2:
```

When the teach pendant is disabled during background editing

If ”-BCKEDT-” is selected in the foreground, the foreground enters the program non-selection state when the teach pendant is disabled. (The status line disappears.) So, the program being edited in the background cannot be executed externally.
The background editing can be continued without modification.

Teach pendant : Enable

```
-BCKEDT-        JOINT    10%
BBB
      <<BACKGROUND>>
    1:
    2:
```

→

Teach pendant : Disable

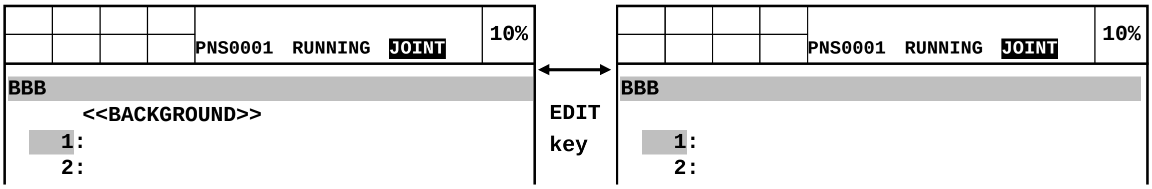
```
                  JOINT    10%
BBB
      <<BACKGROUND>>
    1: J  P[1] 100% FINE
    2:
```

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.

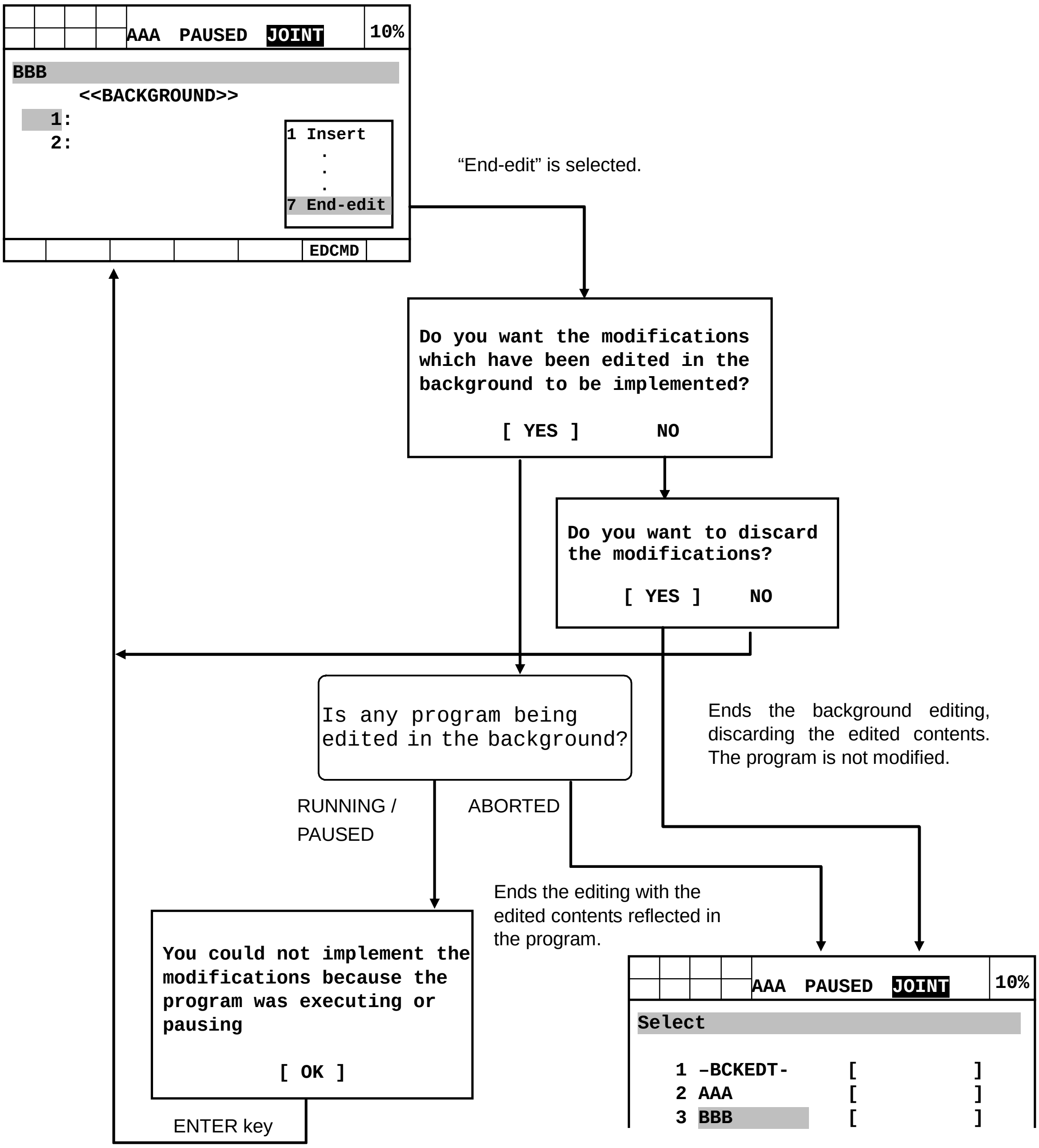
Teach pendant : Enable **Teach pendant : Disable**



If no program is selected in the foreground, pressing the edit key does not switch screen display; the error “Program is not selected” occurs.
If no program is selected for background editing, pressing the edit key does not switch screen display; the error “Not editing background program” occurs.

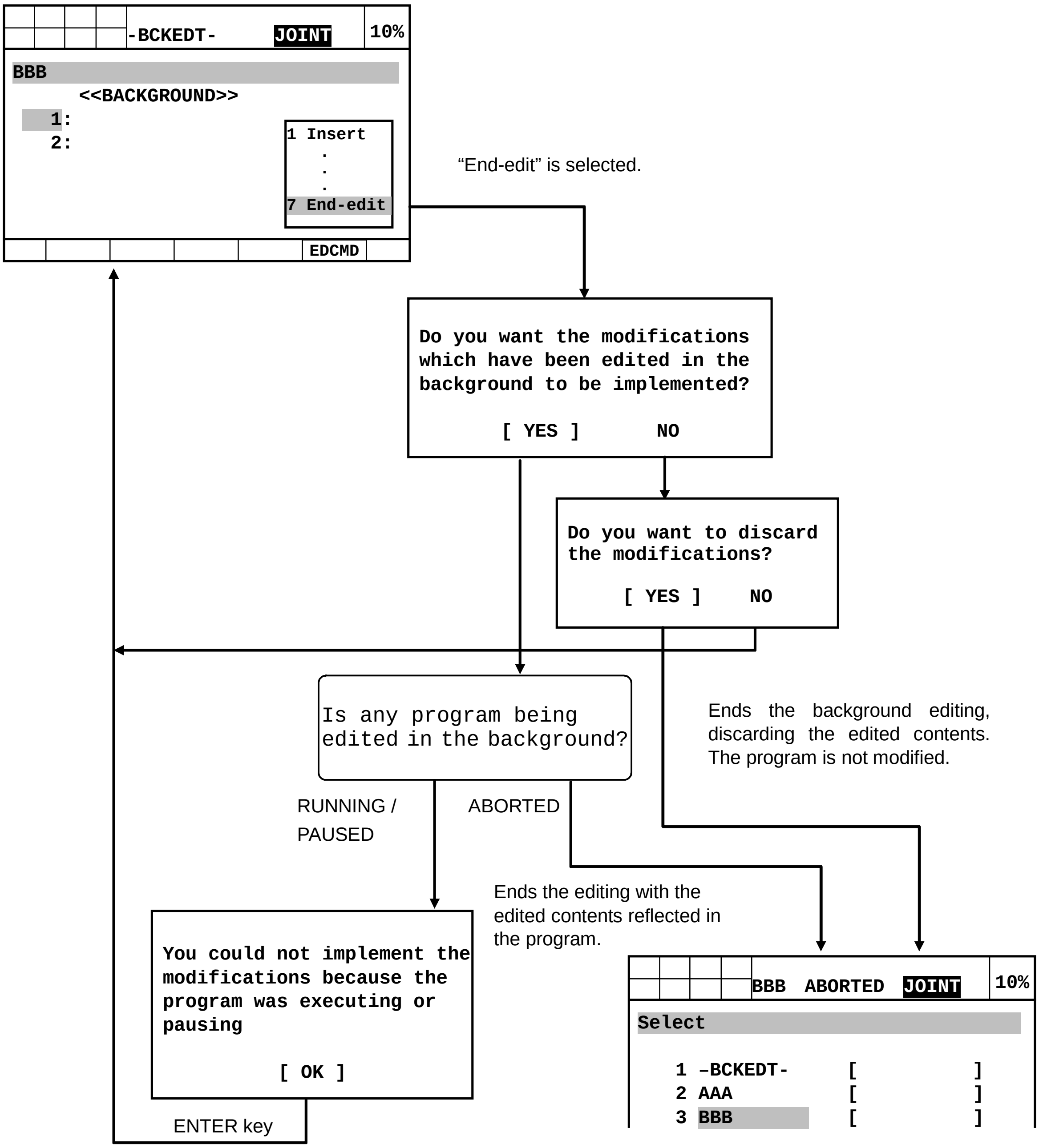
When background editing is terminated with the teach pendant disabled

When background editing is terminated, the program directory screen will be displayed. At this time, the user can specify whether to reflect the results of background editing in the original program.



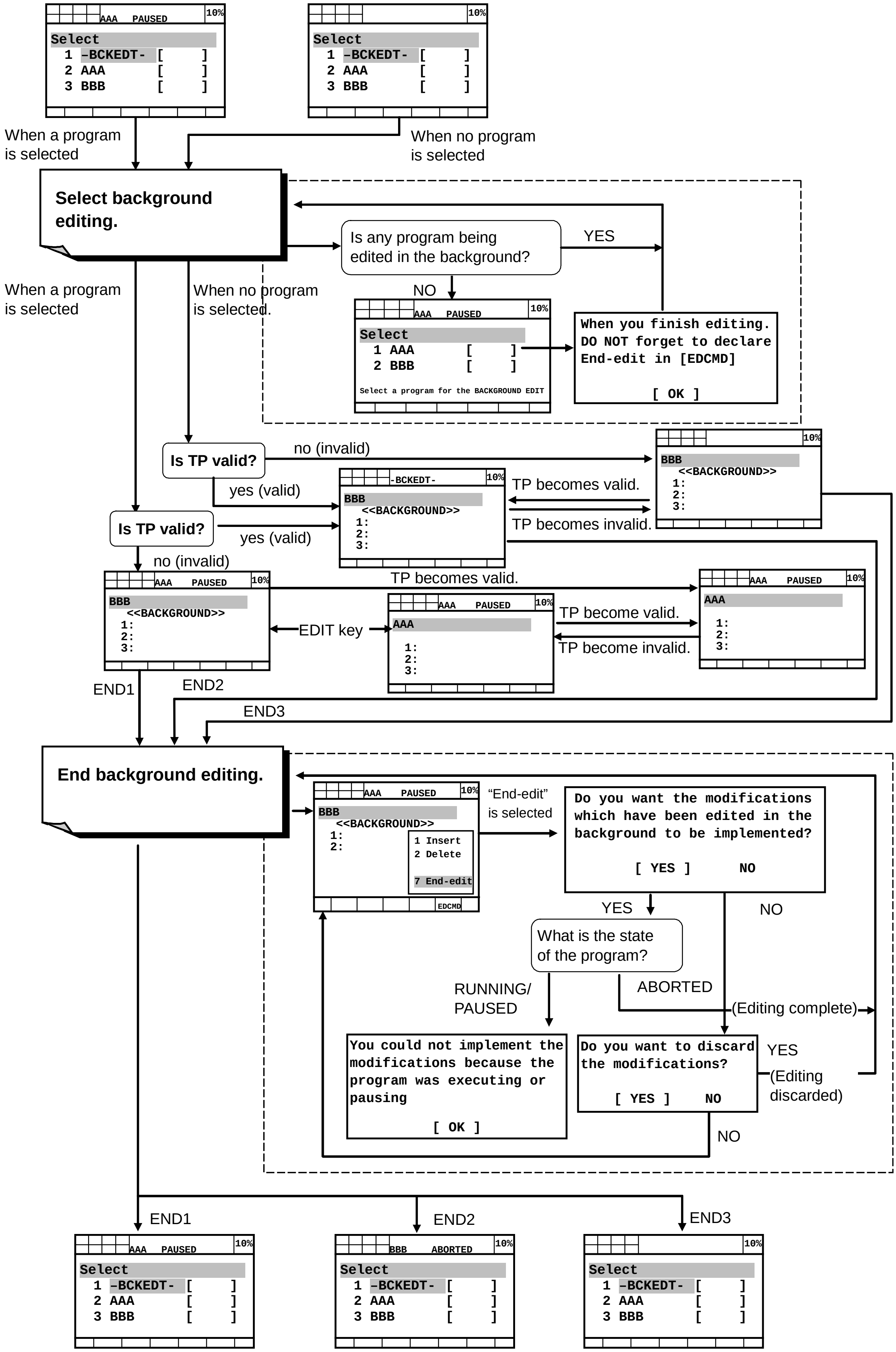
When background editing is terminated with the teach pendant enabled

When background editing is terminated, the program directory screen will be displayed. 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 10%
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
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 no program is selected.
Is TP valid?
no (invalid)
yes (valid)
-BCKEDT-
BBB
<<BACKGROUND>>
TP becomes valid.
TP becomes invalid.
EDIT key
TP become valid.
TP become invalid.
END1
END2
END3
End background editing.
"End-edit" is selected
1 Insert
2 Delete
7 End-edit
EDCMD
Do you want the modifications which have been edited in the background to be implemented?
[YES] NO
What is the state of the program?
RUNNING/ PAUSED
ABORTED
(Editing complete)
You could not implement the modifications because the program was executing or pausing
[OK]
Do you want to discard the modifications?
[YES] NO
(Editing discarded)
BBB ABORTED

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 is cycled 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, the original program is restored 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.

    The following errors occur, 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.

    The following errors occur, and the robot stops:

    SYST-011 Failed to run task

    PROG-006 Line is not found
- When the original program is write-protected (Write-protect is ON), editing cannot be implemented. In this case, the following errors occur:

  TPIF-054 Could not end editing

  TPIF-008 Memory protect violation

  Please cancel to implement modification to terminate Background editing.
- Background editing can be terminated even when the special program for background editing is write-protected.

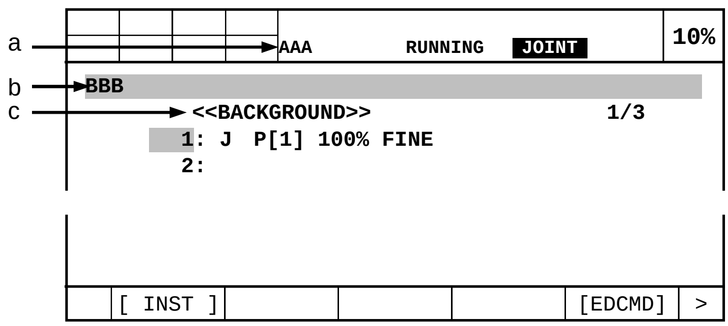
- The status line displays the execution state of a selected program. So, if a subprogram being executed is terminated forcibly, and the main program is selected in the foreground, the status line continues to display the subprogram name.
  If program start operation is initiated here, the execution 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 will be displayed.
- When there is a suspended program in the background, the special program for background editing (”-BCKEDT-”) cannot be read from the external storage device. In this case, the following message will be displayed:
  This program is being edited
  Before reading the special program from the external storage device, terminate background editing.
- For Foreground program, editor can be switched to the called programs with the [ENTER] key on program name of CALL instruction. But for background program, it cannot be switched. Please terminate background edit and select called programs.
- When using background editing function, the program is switched between selected and displayed on the edit screen depending on the status of the teach pendant enable/disable.
  Notes about switching this selection are as follows.
  - The teach pendant is enabled, and the program being edited in the background is selected from the program select screen, the paused program is aborted and the selected program is cleared. So, the paused program cannot be restarted.
  - When background editing is continued after the teach pendant changed to enabled or when background editing is finished, this above operation is needed. So the paused program cannot be restarted.
  - When you need to restart the paused program, please continue the background editing or finish background editing with the teach pendant disabled.

The operation flow is explained below.

The teach pendant is disabled (Program being executed)

During program AAA being executed, program BBB is edited in the background.

1) Select “-BACKEDT-”, and select program BBB for editing.
2) The teach pendant screen is changed to the background editing screen of program BBB as below.



a : Program name selected in the foreground.

b : Program name selected in the background.
c : Indication of background editing state.

This state is that program AAA is selected in the foreground and program BBB is selected in the background.

Teach pendant is enabled.

Note the operation after the teach pendant is enabled from above state.

3) With program AAA selected in the foreground and program BBB selected in the background, when the teach pendant is enabled, the teach pendant edit screen is changed from program BBB being edited in the background to program AAA selected in the foreground automatically.

   The purpose of this automatic changing screen is that program AAA being executed in the foreground can be operated by changing the teach pendant enabled even though the other program being edited in the background.

Screen of program BBB being edited in the background

```
            AAA         RUNNING  JOINT        10%
BBB
          <<BACKGROUND>>                  1/3
       1: J  P[1] 100% FINE
       2:

  [ INST ]                       [EDCMD]  >
```

The teach pendant is enabled.

Program AAA selected in the foreground is displayed automatically.

```
            AAA         PAUSED  JOINT         10%
AAA
                                          1/3
       1: L  P[1] 1000mm/sec FINE
       2:

  [ INST ]                       [EDCMD]  >
```

4) Next, select “-BCKEDT-” from the program select screen to continue the background editing of program BBB.

***At this moment, note the following.**

If select “-BCKEDT-” with the teach pendant enabled, the program in the foreground is changed from program AAA to program –BCKEDT-. Namely, program AAA is changed to non-selected program at this point, and execution status is changed from paused to aborted simultaneously.
(This behavior is equal to select another program while a paused program exists.)
Therefore if background editing of program BBB is finished afterward, the edit screen of program AAA is not displayed.
Program AAA need to be selected from the program select screen to edit or execute it. However if program AAA is selected, it cannot be restarted at paused line.

The edit screen of program AAA selected in the foreground

```
AAA                    PAUSED JOINT      10%
AAA
                                        1/3
     1: L  P[1] 1000mm/sec FINE
     2:

[ INST ]                      [ EDCMD ]  >
```

Select "-BCKEDT-".

The background editig screen of program BBB is displayed.

```
-BCKEDT-               ABORTED JOINT     10%
BBB
        <<BACKGROUND>>                  1/3
     1: J  P[1] 100% FINE
     2:

[ INST ]                      [EDCMD]    >
```

Please select "-BCKEDT-" after teach pendant is disabled, to continue the background editing without program AAA is changed to non-selected program. In that case the program selected in the foreground remains as program AAA as shown below.

```
AAA                    PAUSED JOINT      10%
AAA
                                        1/3
     1: L  P[1] 1000mm/sec FINE
     2:

[ INST ]                      [EDCMD]    >
```

<u>After the teach pendant is disabled,</u>
select "-BCKEDT-".

The background editing screen of program BBB is displayed.

```
AAA                    PAUSED JOINT      10%
BBB
        <<BACKGROUND>>                  1/3
     1: J  P[1] 100% FINE
     2:

[ INST ]                      [EDCMD]    >
```

Background Look Function

For Background Look function, programs can be looked and confirmed on the screen even if any programs are running.

Programs cannot be edited in Background Look mood. Please open the program as Background Edit mode if editing is needed.

Background Look function is started by selecting –BCKEDIT- on selection screen. "<<BACKGOUND LOOK>>" is displayed on top of the editor screen when the program is opened as Background Look mode.

```
PROG001
        <<BACKGROUND LOOK>>                1/10
           1:   R[1]=DI[2]
           2:   DO[3]=ON
           3:   R[R[1]]=DI[R[2]]
           4:   PR[1]=P[3]
           5:   PR[1,2]=PR[R[3],R[4]]
           6:   PL[1]=PL[R[3]]
           7: J  PR[1] 100% FINE
           8: J  P[1] 100% FINE
           9:   LBL[1]
        [End]

   END_LOOK                           [EDCMD]   >
```

Procedure 5-39 Open program on Background Look

Step

1 Select SELECT. The program selection screen is displayed.

```
Select
                    61092 bytes free      9/10
      No. Program name         Comment
       1  -BCKEDT-        [                  ]
       2  GETDATA      MR [Get PC Data       ]
       3  REQMENU      MR [Request PC Menu   ]
       4  SENDDATA     MR [Send PC Data      ]
       5  SENDEVNT     MR [Send PC Event     ]
       6  SENDSYSV     MR [Send PC Sysvar    ]
       7  SAMPLE1         [SAMPLE PROGRAM 1  ]
       8  SAMPLE2         [SAMPLE PROGRAM 2  ]
       9  PROG001         [PROGRAM001        ]
      10  PROG002         [PROGRAM002        ]

  [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR ]  >
```

2 Select –BACKEDT-.

```
Select
                    61092 bytes free      8/9
      No. Program name         Comment
       2  GETDATA      MR [Get PC Data       ]
       3  REQMENU      MR [Request PC Menu   ]
       4  SENDDATA     MR [Send PC Data      ]
       5  SENDEVNT     MR [Send PC Event     ]
       6  SENDSYSV     MR [Send PC Sysvar    ]
       7  SAMPLE1         [SAMPLE PROGRAM 1  ]
       8  SAMPLE2         [SAMPLE PROGRAM 2  ]
       9  PROG001         [PROGRAM001        ]
      10  PROG002         [PROGRAM002        ]

    Select a program for the BACKGROUND EDIT

                                   LOOK    EDIT
```

3 Move the cursor to the name of a program to display, and then press the F4 LOOK key.
Program is displayed on Background Look screen.
(If F5 EDIT is pushed, program is opened as Background Edit mode.)

```
PROG001
        <<BACKGROUND LOOK>>                 1/10
        1:   R[1]=DI[2]
        2:   DO[3]=ON
        3:   R[R[1]]=DI[R[2]]
        4:   PR[1]=P[3]
        5:   PR[1,2]=PR[R[3],R[4]]
        6:   PL[1]=PL[R[3]]
        7: J  PR[1] 100% FINE
        8: J  P[1] 100% FINE
        9:   LBL[1]
     [End]

   END_LOOK                          [EDCMD]   >
```

4 To close Background Look, press the F1 END_LOOK key, or select [EDMD] and End_edit.

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 Subsection 4.3.2.)
To prevent such trouble, the singular point check function checks to see if a taught position is a singular point when the position is taught. Then, the function teaches such a position according to 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 registered position type is rectangular type.
- 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.

If a check finds that the taught position is a singular point, the top two lines of the teach pendant display the following warning message:

TPIF-060 Can’t record on Cartesian (G:i)
MOTN-023 In singularity
i: Move group number at a singular point

At the same time, the following prompt message is displayed at the lower part of the teach pendant:

Record current position on joint

At this time, the function keys YES and NO are displayed. Select one of the two keys.

- YES: Registers 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.

> **NOTE**
> This function is not applicable to the teaching of typical palletizing loading points and passing points.

5.8 OTHER EDITING FUNCTION

This section describes following functions.

- Auto position renumbering
- Fixed program name
- Filtered program list

5.8.1 Auto Position Renumbering

This function renumbers Position number automatically when:

- New motion statement is taught by SHIFT + F1, POINT.
- New position is taught by SHIFT+F5, TOUCHUP.
- Program line that includes position is deleted
- Program line that includes position is copied and pasted

Function

This function is disabled in default setting.
To enable this function, set $POS_EDIT.$AUTO_RENUM2 to be TRUE. (Default: FALSE)

(Example) Teach new position to the 3rd line by SHIFT + F1, POINT.

```
PNS0005
                                          3/6
        1: J  P[1] 100% FINE
        2: J  P[2] 100% FINE
        3:
        4: J  P[3] 100% FINE
        5: J  P[4] 100% FINE
     [End]

   | POINT |  |  |  | TOUCHUP | >
```

**Case that this function is enabled:
Position number is renumbered automatically.**

```
PNS0005
                                          3/6
        1: J  P[1] 100% FINE
        2: J  P[2] 100% FINE
        3: J @P[3] 100% FINE
        4: J  P[4] 100% FINE
        5: J  P[5] 100% FINE
     [End]

   | POINT |  |  |  | TOUCHUP | >
```

Case that this function is disabled (default):

```
PNS0005
                                          3/6
        1: J  P[1] 100% FINE
        2: J  P[2] 100% FINE
        3: J @P[5] 100% FINE
        4: J  P[3] 100% FINE
        5: J  P[4] 100% FINE
     [End]

   | POINT |  |  |  | TOUCHUP | >
```

5.8.2 Fixed Program Name

This function restricts the name of program. Only the program name that starts from registered word become valid. Registered word for the program can be customized in system configuration menu. (Refer to 3.15 System Config Menu) The program that does not start from registered word cannot be created. The alarm “TPIF-038 Invalid char in program name” occurs when the invalid program name is specified.

Function

This function is disabled in default setting.
To enable this function, set $ PGINP_PGCHK to be 1. (Default: 0)

```
System/Config
                                          17/47
     14 Wait timeout:              30.00 sec
     15 Receive timeout:           30.00 sec
     16 Return to top of program:    TRUE
     17 Original program name(F1): [RSR     ]
     18 Original program name(F2): [PNS     ]
     19 Original program name(F3): [STYLE   ]
     20 Original program name(F4): [JOB     ]
     21 Original program name(F5): [TEST    ]
     22 Default logical command: <*DETAIL*>
     23 Maximum of ACC instruction    150
     24 Minimum of ACC instruction      0

   [ TYPE ]
```

In this setting, only the programs that begins from RSR, PNS, STYLE, JOB, TEST become valid.

5.8.3 Filtered Program List

This function filters program list displayed on program list screen. Only the programs that start from registered words become displayed on program list screen.

5.8.3.1 Method to use the original program name specified in the system configuration menu as the head of the string

Registered word for the program can be customized on system configuration menu. (Refer to 3.15 System Config Menu.)

Function

This function is disabled in default setting.
To enable this function, set $ PGINP_FLTR to be 1 or 2. (Default: 0)

1. Press [TYPE] key on program list screen.

Case $PGINP_FLTR = 1:
Program type and registered words are displayed as choices.

```
Select
                 61092 bytes free      1/22
      No. Program name         Comment
         TYPE  1 DT-      [                  ]
 1 ALL          TA     MR [Get PC Data       ]
 2 TP Programs  NU     MR [Request PC Menu   ]
 3 Macro        ATA    MR [Send PC Data      ]
 4 Cond         VNT    MR [Send PC Event     ]
 5 ‘RSR’        YSV    MR [Send PC Sysvar    ]
 6 ‘PNS’        E1        [SAMPLE PROGRAM 1  ]
 7 ‘STYLE’      E2        [SAMPLE PROGRAM 2  ]
 8 ‘JOB’        01        [PROGRAM001        ]
 9 ‘TEST’       02        [PROGRAM002        ]
   TYPE   CREATE   DELETE   MONITOR  [ATTR ]  >
```

Case $PGINP_FLTR = 2:
Registered words are displayed as choices.

```
Select
                 61092 bytes free      1/22
      No. Program name         Comment
        1  -BCKEDT-       [                  ]
        2  GETDATA     MR [Get PC Data       ]
        3  REQMENU     MR [Request PC Menu   ]
    TYPE    1  ATA     MR [Send PC Data      ]
 1 ALL         VNT     MR [Send PC Event     ]
 2 RSR         YSV     MR [Send PC Sysvar    ]
 3 PNS         E1         [SAMPLE PROGRAM 1  ]
 4 STYLE       E2         [SAMPLE PROGRAM 2  ]
 5 JOB         01         [PROGRAM001        ]
 6 TEST        02         [PROGRAM002        ]
   TYPE   CREATE   DELETE   MONITOR  [ATTR ]  >
```

2. Select “PNS”. Only the programs that start from selected word are displayed.

```
Select
                 61092 bytes free      1/4
      No. Program name         Comment
        1  -BCKEDT-       [                  ]
        2  PNS0001        [                  ]
        3  PNS0002        [                  ]
        4  PNS0003        [                  ]
        5                 [                  ]
        6                 [                  ]
        7                 [                  ]
        8                 [                  ]
        9                 [                  ]
       10                 [                  ]
   [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR ]  >
```

5.8.3.2 Method to register 6 or more strings as the head of string

With the method “5.8.3.1 Method to use original program name specified in the system configuration menu as the head of the string”, only 5 words can be registered as the head of string of program list. Up to 30 strings can be registered as the head of string by the following method.

Procedure 5-40 Filtering by registering more than 6 strings as the head of string

1. Set the string that you want to register as the head of string.

- Select “MENU → 0.NEXT → 6.SYSTEM → Variables” so that the system variables list screen will be displayed.
- On the system variables list screen, set the string that you want to register as the head of string to the system variable $PGINP_TYPE. In this example, set the system variable $PGINP_TYPE as follows.
  $PGINP_TYPE[1] = ‘ABC’
  $PGINP_TYPE[2] = ‘DEF’
  $PGINP_TYPE[3] = ‘GHI’
  $PGINP_TYPE[4] = ‘JKL’
  $PGINP_TYPE[5] = ‘MNO’

2. Set the system variable $PGINT_FLTR.

- On the system variable list screen, set the system variable $PGINT_FLTR.

- In default setting, the value of the system variable $PGINT_FLTR is 0. In this case, only the list of program type, PROGRAM, MACRO, etc. is displayed in the list displayed by F1, [TYPE] key on the program list screen.
- Set the system variable $PLINT_FLTR to 5 or 6. The head of the string that registered in the system variable $PGINP_TYPE can be selected on the list displayed by F1, [TYPE] key on the program list screen.

Case $PGINP_FLTR = 5:
Program type and registered words are displayed as choices.

```
Select
                 61092 bytes free      1/22
      No. Program name         Comment
          -BCKEDT-     [               ]
   TYPE   1  A       MR [Get PC Data      ]
   1 ALL     U       MR [Request PC Menu  ]
   2 TP Programs TA  MR [Send PC Data     ]
   3 Macro   NT      MR [Send PC Event    ]
   4 Cond    SV      MR [Send PC Sysvar   ]
   5 'ABC'   1          [SAMPLE PROGRAM 1 ]
   6 'DEF'   2          [SAMPLE PROGRAM 2 ]
   7 'GHI'   1          [PROGRAM001       ]
   8 'JKL'   2          [PROGRAM002       ]
   9 'MNO'
     TYPE   CREATE   DELETE   MONITOR   [ATTR ]   >
```

Case $PGINP_FLTR = 6:
Registered words are displayed as choices.

```
Select
                 61092 bytes free      1/22
      No. Program name         Comment
       1  -BCKEDT-       [                ]
       2  GETDATA     MR [Get PC Data      ]
       3  REQMENU     MR [Request PC Menu  ]
   TYPE      1     TA MR [Send PC Data     ]
   1 ALL           NT MR [Send PC Event    ]
   2 ABC           SV MR [Send PC Sysvar   ]
   3 DEF           1     [SAMPLE PROGRAM 1 ]
   4 GHI           2     [SAMPLE PROGRAM 2 ]
   5 JKL           1     [PROGRAM001       ]
   6 MNO           2     [PROGRAM002       ]
     TYPE   CREATE   DELETE   MONITOR   [ATTR ]   >
```

3. <u>Select the registered string from the list displayed by F1, [TYPE] key on the program list screen.</u>

Select "ABC" on the list displayed by F1, [TYPE] key on the program list. Only the programs that have the string "ABC" in the head of program name are listed as follows. Up to 30 strings can be registered to the system variable "PGINP_TYPE". Register the string that you often use as the head of program name to filter the program name on the program list screen.

```
Select
                 61092 bytes free      1/4
      No. Program name         Comment
       1  -BCKEDT-     [                 ]
       2  ABC0001      [                 ]
       3  ABC0002      [                 ]
       4  ABC0003      [                 ]
       5               [                 ]
       6               [                 ]
       7               [                 ]
       8               [                 ]
       9               [                 ]
      10               [                 ]
   [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR ]  >
```

5.8.4 Program Collection Function

There are cases that it is difficult to search for the objective robot application program on the program Select menu when there are too many robot application programs. In this case, it is possible to separate the programs to categories and coordinate the programs by using this function.

"collection" is program item that collect and display only programs, which are registered to the collection on the program Select menu. And those collection programs do not function as robot application programs. And also, it is expected to enhance operability by using this function along with using Tree View function. (refer to "11.1.6 Tree View")

5.8.4.1 Setup collection

- **$COLLECT_ENB:**

0: Colleciton is disabled.
1 (init value): Collection is enabled

- **$COLLECT_CFG.$MULTI_PROG:**

FALSE: One robot application program can be registered to only one collection.
TRUE (init value): One robot application program can be registered to multi collections.

- **$COLLECT_CFG.$ALLOW_PROC:**

FALSE (init value): Programs whose subtype is process cannot be registered to collection.
TRUE: Programs whose subtype is process can also be registered to collection.

5.8.4.2 How to create and use collections

A procedure how to register programs as collection is written on the following “Procedure 5-40 Creating collections”, a procedure how to add a program to a collection on the following “Procedure 5-41 Adding a program to collection”, and a procedure how to use collections on the following “Procedure 5-42 How to use collections”.

> **NOTE**
> Collection program function is available on system software version 7DC3 series or later and *i*Pendant firmware version 7DC3 series or later.

Procedure 5-41 Creating collections

Step

1. Register a program (refer to “Procedure 5-2 Registering a program”). At this time, don’t teach the program on EDIT screen).
2. Suit key cursor to the program on program Select menu and press F7[DETAIL] and open program detail screen.

```
Program detail
                                                     1/7
      Creation Date:           16-Jan-1994
      Modification Date:       08-Mar-1994
      Copy Source:

      Positions: FALSE  Size:      190 Bytes

       Program name:
       1 LINE1
       2  Sub Type:      [None              ]
       3  Comment:       [                  ]
       4  Group Mask:    [1,*,*,*,*,*,*,*   ]
       5  Write protect: [OFF               ]
       6  Ignore pause:  [OFF               ]
       7  Stack size:    [     500          ]
       8  Collection:    [                  ]

   END      PREV     NEXT
```

3. Make Sub Type be Collection. (It is impossible to make sub type be collection if there are any instruction(s) on EDIT screen)

```
Program detail
                                           1/7
      Creation Date:          16-Jan-1994
      Modification Date:      08-Mar-1994
      Copy Source:

      Positions: FALSE  Size:      190 Bytes

       Program name:
       1 LINE1
       2  Sub Type:      [Collection      ]
       3  Comment:       [                ]
       4  Group Mask:    [1,*,*,*,*,*,*,* ]
       5  Write protect: [OFF             ]
       6  Ignore pause:  [OFF             ]
       7  Stack size:    [     500        ]
       8  Collection:    [                ]

    END      PREV      NEXT
```

4 Suit the key cursor to the item Collection and press the [ENTER] key so that Collection Editor screen is displayed.

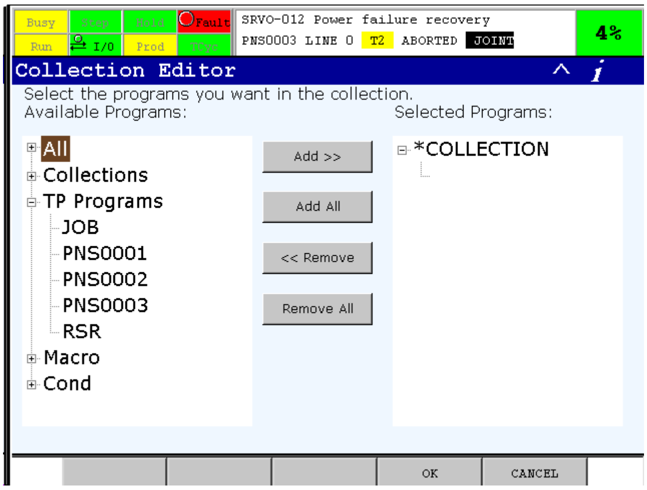


Fig. 5.8.4 (a) Collection Editor Screen 1

5 Available programs as being registered to the collection are displayed as Tree View on the left side of the screen. Select a program and press Add key so that the program is added to the collection. Added programs are displayed on the right side of the screen. Select a program and press Remove key if you want to cancel a program. And also, it is possible to add all registrable programs when you press Add All key and it is possible to cancel all registered programs when you press Remove All key.

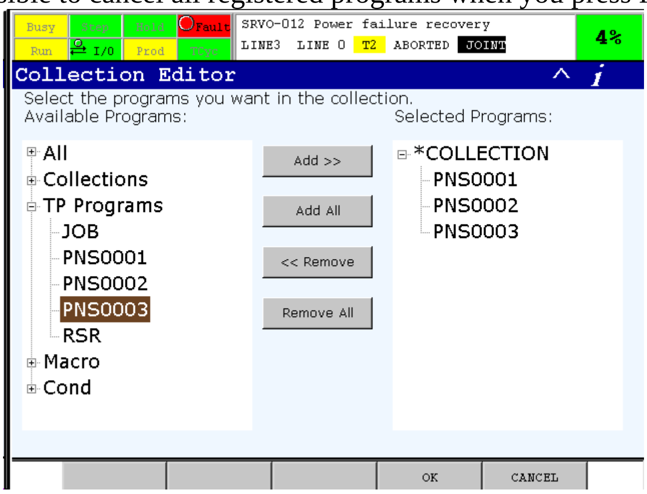


Fig. 5.8.4 (b) Collection Editor Screen 2

NOTE
If your *i*Pendant is not support touch panel, You can move focus to other elements by pressing *i* key + Down key. (This operation is corresponding to Tab key on PC operation)

6 Press F4[OK] so that the registration is complete after you selected all programs that you want to make those programs be registered to the collection.

Procedure 5-42 Adding a program to a collection

Condition

- An objective collection program has been created.

Step

1 Suit key cursor to the objective program to be registered to the collection program and press F7[DETAIL] so that Program Detail screen is displayed.
2 Suit key cursor to the item Collection and press the [ENTER] key so that Collection List screen is displayed.

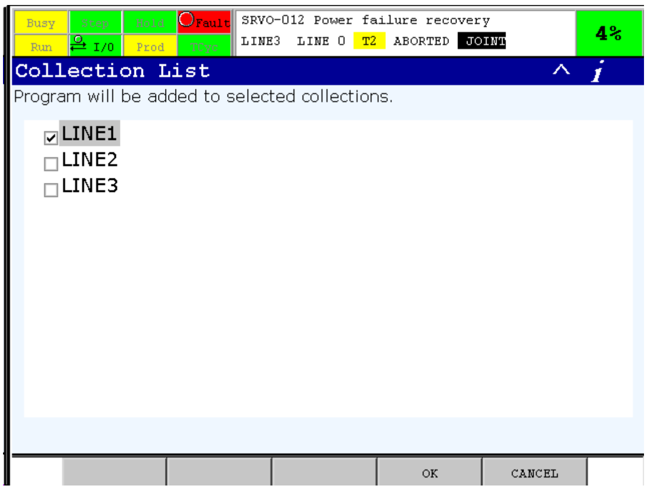


Fig. 5.8.4 (c) Collection List Screen

3 Select the objective collection program and press F4[OK].

NOTE
When you create TP programs on the condition that a collection is selected(only programs registered to the collection are displayed), the created TP programs will be registered to the collection automatically

Procedure 5-43 How to use collection programs

Condition

- An objective collection program has been made.
  (refer to “5-41 Creating collections”, “5-42 Adding a program to a collection”)

Step

1 Press F1[TYPE] and select Collections on Program Select screen.

```
Select
                  61276 bytes free      1/10
      No. Program name         Comment
       1  -BCKEDT-           [                  ]
       2  GETDATA        MR [Get PC Data        ]
       3  REQMENU        MR [Request PC Menu    ]
       4  SENDDATA       MR [Send PC Data       ]
   TYPE     1            MR [Send PC Event      ]
   1 All                 MR [Send PC Sysvar     ]
   2 Collections            [SAMPLE PROGRAM1    ]
   3 TP programs            [SAMPLE PROGRAM2    ]
   4 Macro                  [PROGRAM001         ]
   5 Cond                   [PROGRAM002         ]
   [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR ]  >
```

2 Select the objective collection.

```
Select
                  61276 bytes free        1/3
      No. Program name    Comment
       1  LINE1        CO [First lane        ]
       2  LINE2        CO [Second lane       ]
       3  LINE3        CO [Third lane        ]

   [ TYPE ]  CREATE  DELETE  MONITOR  [ ATTR]  >
```

3 The collection program and only the registered programs to the collection are listed on Program Select screen.

```
Select
                  61276 bytes free        1/3
      No. Program name    Comment
       1  -BAKED-         [                  ]
       2  LINE1        CO [First lane        ]
       3  PNS0001         [                  ]
       4  HOME_POS        [                  ]
       5  SUB001          [                  ]
       6  SUB002          [                  ]
       7  SUB003          [                  ]
       8  SUB004          [                  ]

   [ TYPE ]  CREATE  DELETE  MONITOR  [ ATTR]  >
```

5.8.4.3 Backup and restore collections

Collections don’t function as robot application program. But collections are backup as program file extension will be .TP). Information for registered program names are recorded when the collections are backup, and programs having the same name are registered to the collections when the collections .

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 decelerated 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 0  T1 ABORTED JOINT   30%
SAMPLE1
```

- 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  T1 PAUSED JOINT   30%
SAMPLE1
```

  To start from another line in the same program or another program, abort a program to release the paused state.

There are following methods to halt a program intentionally:

- Press the emergency stop button on the teach pendant or the machine operator’s panel or release the deadman switch.
  Peripheral device I/O *IMSTP input
- Press the HOLD button on the teach pendant or use the input signal *HOLD of the peripheral I/O:

These inputs halt the execution of the program.

- Select 1 ABORT(ALL) from the function menu. Peripheral device I/O *CSTOPI input.

This method aborts the program.

6.1.1 Halt by an Emergency Stop and Recovery

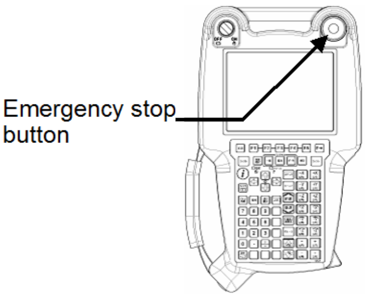
To stop the robot immediately, press the emergency stop button on the machine operator’s panel/box or teach pendant (Please refer to "STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS for detail of stop type). When this is done an emergency stop alarm occurs.

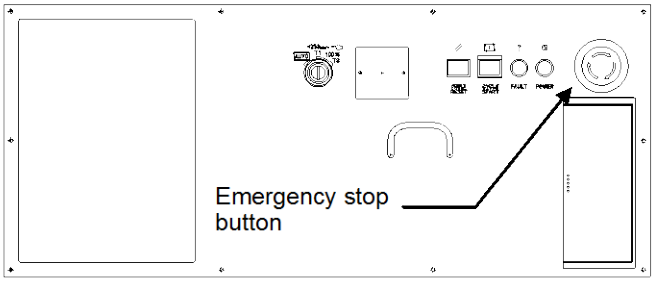
Procedure 6-1 Emergency stop and recovery

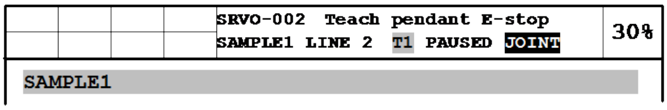
Emergency stop procedure

- Step

1 Press the emergency stop button on the teach pendant or the machine operator’s panel. This halts the running program, PAUSED is displayed on the teach pendant.
The emergency stop button is locked to keep it pressed (on state). The emergency stop alarm message is displayed on the screen of the teach pendant. The FAULT lamp lights.

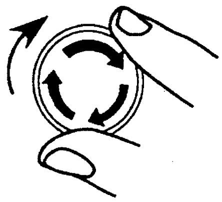




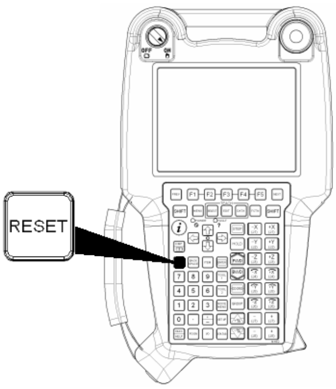


Recovery procedure

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 (or operator’s panel). 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 [HOLD] key causes the following:

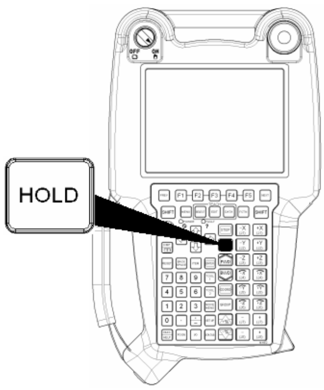
- The robot decelerates slowly until it stops (the program is halted).
- A setting can be made to cause an alarm to turn off the servo power. To make this setting, select SETUP General on the general item setting screen. (→ See Section 3.16, "SETTING THE GENERAL ITEMS".)

Procedure 6-2 Hold and recovery

Hold procedure

- Step

1 Press [HOLD] key on the teach pendant. The running program will be halted, and PAUSED will be 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 will be aborted then the halt state is released.

6.1.3 Halt Caused by an Alarm

An alarm is issued when a failure is detected or when the emergency stop signal or another alarm signal is input from a peripheral device while the operator teaches or plays back a program. When an alarm is generated, it is indicated on the teach pendant, and processing such as robot operation and program execution is stopped to ensure safety.

Displaying an alarm

The operator can check whether an alarm has occurred by watching the FAULT lamps on the teach pendant or the operator's panel and the first line and second line on the screen of the teach pendant.

The kind of an alarm is recognized by an alarm code. The cause and corrective action of an alarm can be known by an alarm code. This manual does not explain the detail of alarm code. Refer to the “FANUC Robot series OPERATOR'S MANUAL (Alarm Code List)” (B-83284EN-1) to know the detail.

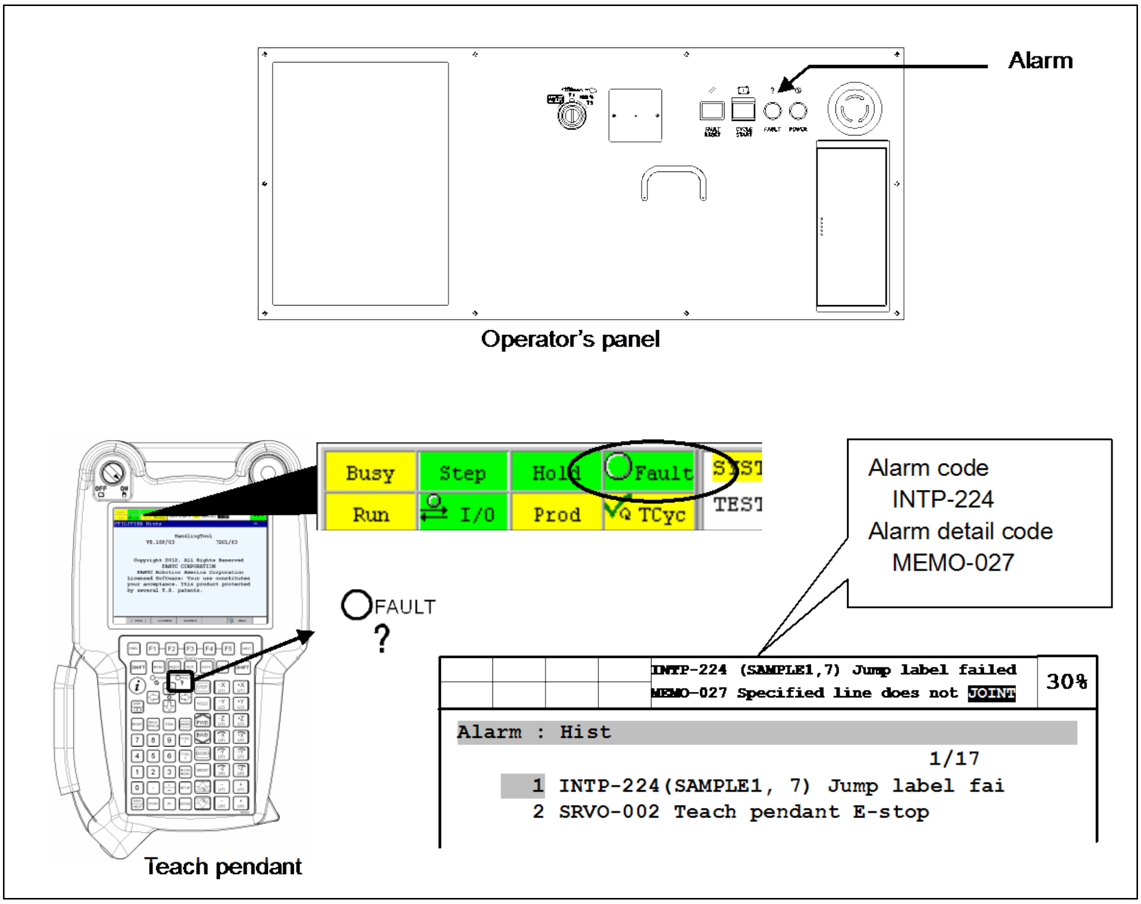


Fig. 6.1.3 (a) Display and indication of an alarm

Alarm history

To display the alarm history, select an alarm history screen [4 ALARM].

```
Alarm : Hist
    DETAIL Alarm
    INTP-224 (SAMPLE1, 7) Jump label failed
    MEMO-027 Specified line does not exist
    STOP.L     01-JUN-98 12:16
    Alarm : Hist
      1 INTP-224 (SAMPLE1, 7) Jump label failed
      2 R E S E T
      3 SRVO-007 External emergency stop
      4 SRVO-001 Operator panel E-stop
      5 R E S E T
      6 SRVO-001 Operator panel E-stop
      7 SRVO-012 Power failure recovery

   [ TYPE ][ VIEW ] ACTIVE   CLEAR   DETAIL
```

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, DETAIL in the alarm history screen [4 ALARM].

```
Alarm : Hist
    DETAIL Alarm
    INTP-224 (SAMPLE1, 7) Jump label failed
    MEMO-027 Specified line does not exist
    STOP.L     01-JUN-98 12:16
    Alarm : Hist
```

1 Alarm code
2 Alarm detail code
3 Alarm severity, Generation date/time

- Alarm code: Identifies an alarm.
- Alarm detail code: Identifies an alarm detail.
- Alarm severity: Indicates the severity of an alarm.
- Generation date: The generation date of the alarm is indicated.

Resetting an alarm

After eliminating the cause of an alarm, press the RESET key to reset the alarm. The alarm indicated in the first and second lines of the teach pendant disappears. When the servo power is turned off, it is turned on. Resetting an alarm usually enables the robot.

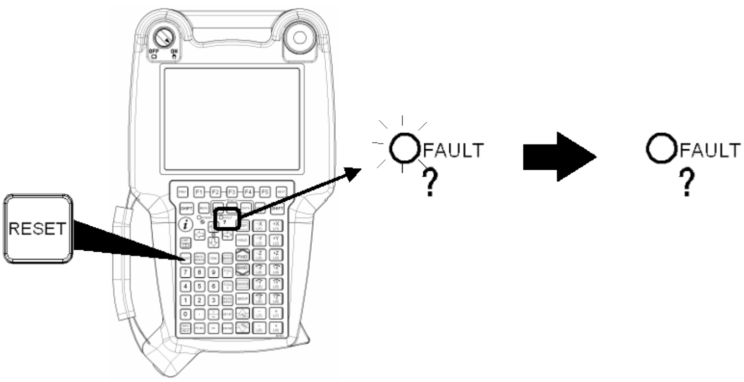


Fig. 6.1.3 (b) RESET key

Disabling the output of peripheral I/O alarm signals

The output of alarm signals (FAULT output) can be disabled.

- Set $ER_NO_ALM.$NOALMENBLE to 1 (enabled).
- Specify the number of alarms for which output is to be disabled in $ER_NO_ALM.$NOALM_NUM.
- Specify the codes of the alarms for which output is to be disabled in $ER_NO_ALM.$ER_CODE1 to $ER_NO_ALM.$ER_CODE10.

11 002 (Meaning: SERVO-002 alarm)
Alarm ID Alarm No.

Halt alarm

The halt alarm function issues an alarm and turns off the power to the servo system when the operator presses the HOLD key to halt the robot.
Specify the fault alarm function in [6 SETUP General] on the general item setting screen (see Section 3.16, “SETTING THE GENERAL ITEMS”).

Alarm severity

The alarm severity indicates the severity of an alarm and the cause of the alarm. Whether program execution and robot operation are stopped, and whether the servo power is turned off depend on the alarm severity.

Table 6.1.3 (a) Alarm severity

	Program	Robot operation	Power to servo system	Range
NONE	none	none	none	—
WARN				—
PAUSE.L PAUSE.G	pause	decelerate the robot slowly until it stops		Local Global
STOP.L STOP.G				Local Global
SERVO		stop the robot immediately	off	Global
ABORT.L ABORT.G	abort	decelerate the robot slowly until it stops	none	Local 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.

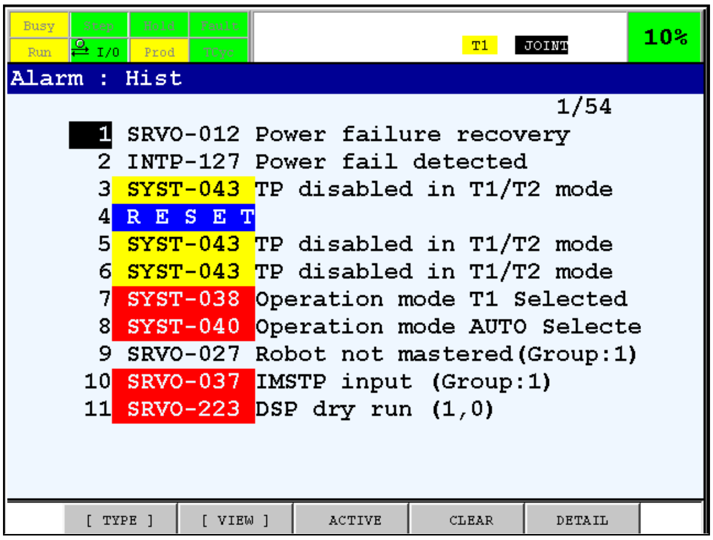
NOTE
Some alarms do not observe the above rules.

Table 6.1.3 (b) Description of alarm severity

Severity	Description
WARN	A WARN alarm warns the operator of a comparatively minor or unimportant failure. The WARN alarm does not affect the operation of the robot. When a WARN alarm occurs, no corresponding LED on the teach pendant or the machine operator's panel lights. To prevent a possible failure in the future, action should be taken for this alarm.
PAUSE	When a PAUSE alarm occurs, the execution of the program is halted, and the operation of the robot is stopped. Appropriate action must be taken for the alarm before the program is restarted.
STOP	When a STOP alarm occurs, the execution of the program is halted, and the robot is decelerated until it is stopped. Appropriate action must be taken for the alarm before the program is restarted.
SERVO	When a SERVO alarm occurs, the execution of a program is paused (or aborted) and the power to the servo system is turned off to stop the robot immediately. The most common cause of a SERVO alarm is hardware failure.
ABORT	When an ABORT alarm occurs, the execution of the program is forcibly terminated, and the robot is decelerated until it is stopped.
SYSTEM	A SYSTEM alarm is issued when a major system failure occurs. When a SYSTEM alarm occurs, every robot in the system is disabled. Contact the FANUC Service Division. After taking appropriate action for the alarm, turn on the power again.

Color display according to the alarm severity

In the alarm display, each alarm code is displayed in the color specified according to its alarm severity.



The color assigned to each alarm severity is listed below:

Table 6.1.3 (c) Color assigned to each alarm severity

Alarm severity	Color
NONE WARN	White
PAUSE.L PAUSE.G	Yellow
STOP.L STOP.G	Yellow
SERVO SERVO2	Red
ABORT.L ABORT.G	Red
SYSTEM	Red
RESET(*)	Blue
SYST-026 System normal power up(*)	Blue

NOTE
Messages "RESET" and "SYST-026 System normal power up" are displayed in blue.

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.
- Using the peripheral device. (RSR 1 to 8 input, PROD_START input, and START input)

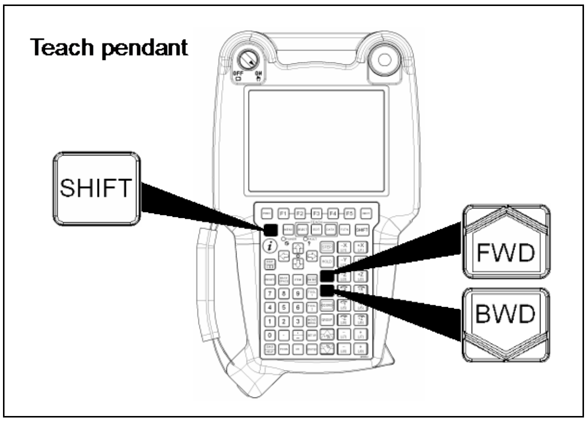


Fig. 6.2.1 (a) Starting a program

For safety's sake, a program can be started only in a device having motion control. Motion control can be switched by using the teach pendant enable switch and the Remote/Local mode change.
(For the Remote/Local mode change, see Section 3.15, "SYSTEM CONFIG MENU".)

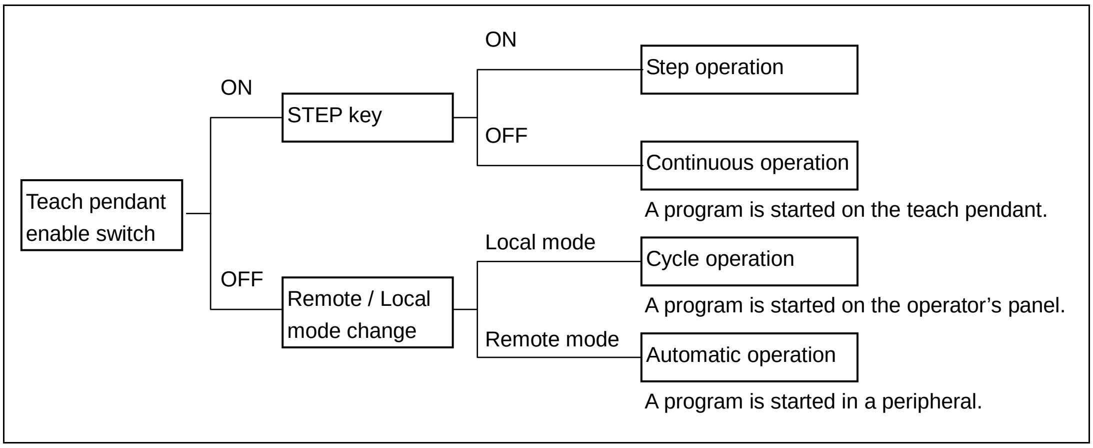


Fig. 6.2.1 (b) How to set the right to start a program

> ⚠ **CAUTION**
> When the start right is switched by using the enable switch on the teach pendant or the Remote/Local mode change, 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:

- Feed rate override : Robot motion speed (operating speed)
- Cartesian coordinate system : Work area where the robot moves

Feed rate override

The feed rate override determines the operating speed. The feed rate override is specified as a percentage of the feed rate specified in the program (programmed speed).
The current feed rate override is displayed in the upper right corner of the screen of the teach pendant, as shown in Fig. 6.2.2 (a).

By pressing the override key, the feed rate override can be changed.

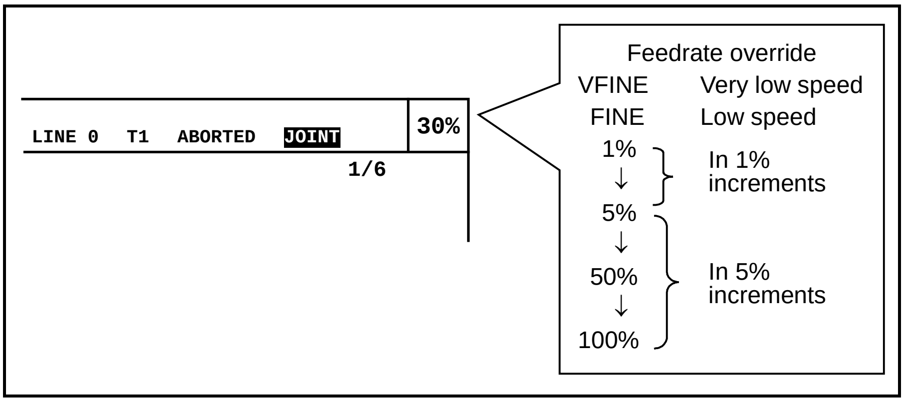


Fig. 6.2.2 (a) Screen display for feed rate override

A feed rate override of 100% would cause the robot to operate at the maximum speed specified in the current setting. Table 6.2.2 shows the change in feed rate override when the override key is pressed. (Table 6.2.2 shows the change in feed rate override in standard setting. The way to change in feed rate override can be customized. Refer to the Subsection 5.2.3.)

Table 6.2.2 Feed rate override (standard setting)

When the override key is pressed	$\text{VFINE} \rightarrow \text{FINE} \rightarrow 1\% \underbrace{\rightarrow}_{\text{In 1\% increments}} 5\% \underbrace{\rightarrow 50\% \rightarrow}_{\text{In 5\% increments}} 100\%$
When the override key is pressed while pressing [SHIFT] key(*1)	VFINE → FINE → 5% → 50% → 100%

*1 Enabled only when $SHFTOV_ENB is 1

To change the feed rate override, press the override key. When the override key is pressed while [SHIFT] key is pressed, the feed rate is changed in the order: VFINE, FINE, 5%, 50%, 100%. However, the feed rate 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 feed rate override of 1%.

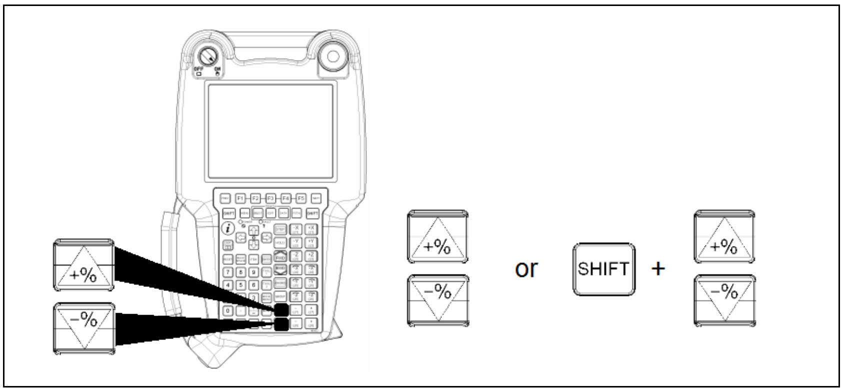


Fig. 6.2.2 (b) Override keys

A feed rate 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 feed rate override.
The feed rate override can only be increased up to the maximum value specified in $SCR.$RUNOVLIM during the program execution. When the feed rate override is over the maximum value, if a program is executed, the feed rate override is decreased to the maximum value.
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 during the program execution.

The system provides a function for allowing the original speed override to be restored when the safety fence is closed. (→ See Section 3.19.)

Operating speed

The operating speed is the speed at which the robot moves while the program is played back. The operating speed is obtained from the following expressions:

Operating speed (joint control motion) (deg/sec, mm/sec) =

$$\text{Maximum joint feed rate} \times \frac{\text{Programmed feedrate}}{100} \times \frac{\text{Feedrate override}}{100}$$

Operating speed (motion under path control) (mm/sec) =

$$\text{Programmed feed rate} \times \frac{\text{Feedrate override}}{100}$$

Operating speed (motion under attitude control) (deg/sec) =

$$\text{Programmed feed rate} \times \frac{\text{Feedrate override}}{100}$$

Fig. 6.2.2 (c) Operating speed

Checking a Cartesian coordinate system

When position data is played back according to Cartesian coordinates, the coordinate system number of the Cartesian coordinate system to be used is checked.
When the specified coordinate system number does not agree with the currently selected coordinate system number, the program is not executed.
The coordinate system number is specified for position data when the position is taught. To change a written coordinate system number, use the tool change function/coordinate system change function.

- Tool coordinate system number (UT)

The number of a mechanical interface coordinate system or tool coordinate system is specified as a tool coordinate system number (UT). This number determines the tool coordinate system.

- 0 : The mechanical interface coordinate system is used.
- 1 to 10 : The tool coordinate system having the specified tool coordinate system number 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 number is used.
- F : The coordinate system having the currently selected user coordinate system number is used.

Position data information

Pressing the F5, POSITION key displays position data information.

```
SAMPLE1
     P[2] UF:0  UT:1        CONF:NUT 000
     X   1500.374   mm  W    40.000  deg
     Y   -242.992   mm  P    10.000  deg
     Z    956.895   mm  R    20.000  deg
    Position Detail
```

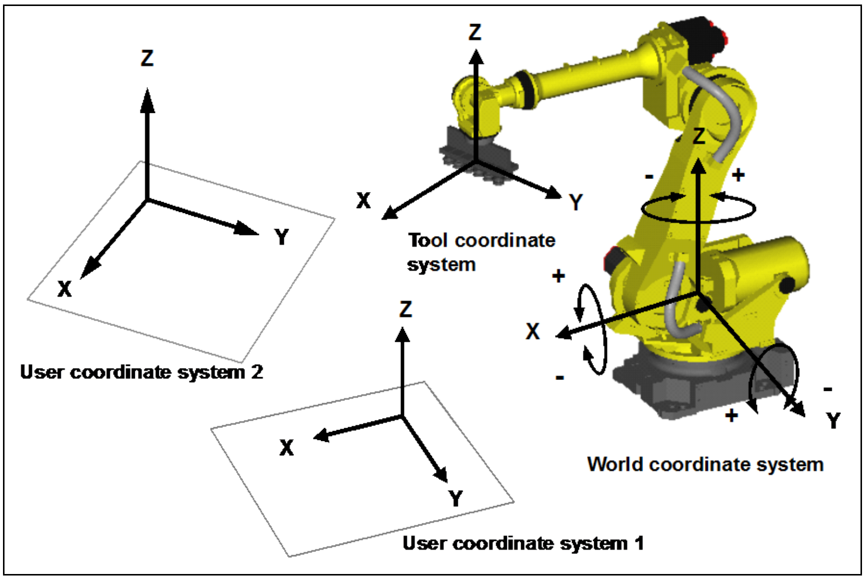


Fig. 6.2.2 (d) Selecting a tool coordinate system and user coordinate system

6.2.3 Resuming a Program

Resuming a program means to restart a halted program. Before a program is halted, the system records the program. As a result, the following is possible:

- Control can be passed to the main program called with the program call instruction.
- The path for a circular motion can be reproduced.
- The path for a circle arc motion can be reproduced. (Refer to the Subsection 9.19 CIRCLE ARC MOTION INSTRUCTION.)

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 Subsection 6.3.2.) The motion is executed at the travel speed specified in the circular motion instruction.

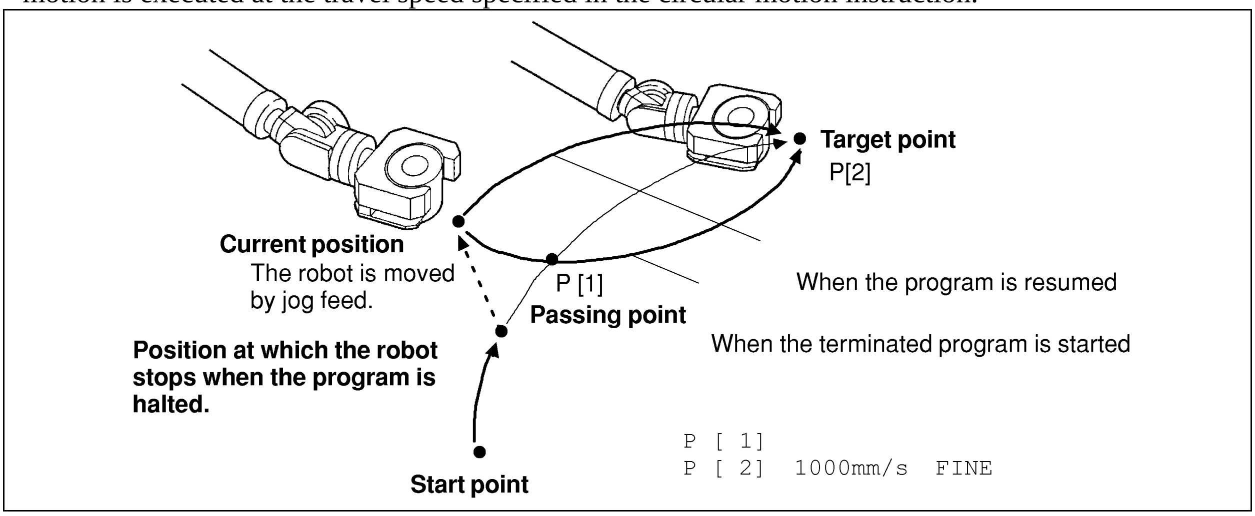


Fig. 6.2.3 Path for a circular motion

Releasing the halt state

The halt state of the program is released when:

- "ABORT (ALL)" is selected from the function menu.
- Switching of the start right. (→ See Subsection 6.2.1.)
- Creating new program when the teach pendant is enabled. For program creation. (→ see Section 5.3.)
- Selecting another program when the teach pendant is enabled. For program selection. (→ see Subsection 5.4.1.)

Moving the cursor in the halt state

When the cursor is moved to a desired line in the halted program and the program is to be resumed, the system asks the operator whether the program is to be resumed at the line to which the cursor has been moved. When YES is selected in response to this message, the program is halted at the line to which the cursor has been moved. When NO is selected, the cursor is returned to the line it was at before it was moved (original line), then the program is halted at that line. For both YES and NO, when the program is resumed, program execution starts at the line to which the cursor has been moved.

Procedure 6-4 Releasing the halt state

Condition

■ The program must be halted. (PAUSED is displayed on the screen.)

```
SAMPLE1  LINE 2  T1  PAUSED  JOINT   30%
SAMPLE1
```

Step

1 Press [FCTN] key to display the function menu.

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

2 Select 1 ABORT (ALL). The program will be terminated. (ABORTED is displayed on the screen.)

```
SAMPLE1  LINE 0  T1  ABORTED  JOINT   30%
SAMPLE1
```

Procedure 6-5 Moving the cursor in the halt state

Condition

■ The program must be halted. (PAUSED is displayed on the screen.)

```
SAMPLE1  LINE 2  T1  PAUSED  JOINT   30%
SAMPLE1
```

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.

```
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.

```
SAMPLE1  LINE 4  T1  PAUSED  JOINT                30%
SAMPLE1
                                          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.

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.
                                              1/6
       1 Group                           : 1
       2 Enable Tolerance checking       : FALSE
       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 the difference of position in one group exceeds the tolerance.
2 Enable Tolerance checking
To enable the restart position check function, select TRUE. (The default setting is TRUE.)
3 Distance Tolerance (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 Orientation Tolerable (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 Axes Tolerance: Rotary 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 Axes Tolerance: Translation axis (mm)
When the difference between the current position of a linear axis in the robot and the position at which the robot was halted is greater than the value set here at program restart, a warning is issued, and the program is not started.

When a program is restarted, this function compares the current robot position with the position at which the robot was halted. If the comparison shows that any of the distance, attitude, and axis position data exceeds a tolerance, a warning is issued, and the program is not started. In this case, the following message appears on the teach pendant:

```
The robot position is out of
stop tolerance.
Please select action.
Choosing CONTINUE will
require cycle start.

  [ STOP ]      CONTINUE
```

(1) When STOP is selected
When “STOP” is selected, this pop-up menu is disappeared, and the program is still paused. After select “STOP”, if start signal input, the tolerance check is executed and the pop-up menu is appeared again.
To resume the program, please move the robot to the position within the tolerance by jog feed, then input start signal.
(2) When CONTINUE is selected
The popup menu disappears, and the program remains halted. When the start signal is input under these circumstances, the program is started.

If jog feed is performed after CONTINUE is selected, checking is made again when the program is restarted next.

> ⚠ **CAUTION**
> This function cannot be used with the tracking function at the same time.

In case that Multi program selection is enabled, while above message is displayed on the teach pendant, other programs cannot be started.

6.3 TESTING

Testing refers to checking the operation of the robot alone before automatically operating the robot in the site line. Testing the program is very important. It must be done to ensure the safety of the workers and the peripheral devices.

The following two methods can be used for testing:

- Step test: Execute the program line by line using the teach pendant or operator's panel.
- Continuous test: Execute the program from the current program line to the end of the program (up to the end-of-program symbol or program end instruction) using the teach pendant or operator's panel.

The teach pendant must be enabled before testing is performed using the teach pendant. The teach pendant is enabled when.

- The teach pendant enable switch is on.

Before test operation can be started from the operator's panel/box, the operator's panel must be in the enabled state. The operator's panel can be placed in this state provided the following conditions are satisfied:

- The enable switch on the teach pendant is set to OFF.
- The system is in local mode. (About switching to the local mode, refer to Remote/Local setup in the Subsection 3.15 SYSTEM CONFIG MENU.)
- 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, and perform step operation from the teach pendant to check program instructions and I/O, and so forth.
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 from the teach pendant at low speed.
4. Perform continuous operation from the operator's panel at high speed and check the position of the robot and the operation timing.

6.3.1 Specifying Test Execution

To specify test execution is to specify the requirements for test execution of a program.

```
TEST CYCLE Setup
                                          1/4
     GROUP:1(    R-2000iB/210F)
       1 Group Motion:          ENABLE
       2 Digital/Analog I/O:    ENABLE
       3 Step statement type:   STATEMENT
       4 Step path node:        OFF









   [ TYPE ]   Group           DISABLE   ENABLE
```

Table 6.3.1 Setting of test execution

Items	Descriptions
Group motion	This function specifies whether the robot is disabled. - DISABLE : The robot is disabled; it ignores all motion instructions. (Robot is locked.) - ENABLE : The robot is enabled, it usually accepts motion instructions. When the robot lock function is ON, the power to the servo system is assumed to be on. Pressing the RESET key resets all the servo alarms. **NOTE** Even when the robot lock is ON, the robot can not be operated when the emergency stop button is pressed.
Digital/Analog I/O	Digital/Analog I/O specifies whether to communicate with a peripheral device via digital I/O 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.
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". (This function is not used currently.)

Procedure 6-6 Specifying test execution

Step

1 Press [MENU] key to display the screen menu.
2 Select TEST CYCLE. The test cycle screen will be displayed.

```
TEST CYCLE Setup
                                                1/4
     GROUP:1(    R-2000iB/210F)
       1 Group Motion:          ENABLE
       2 Digital/Analog I/O:    ENABLE
       3 Step statement type:   STATEMENT
       4 Step path node:        OFF

   [ TYPE ]   Group            DISABLE   ENABLE
```

3 Specify requirements for test execution.
4 To change the group number, press F2, GROUP.

6.3.2 Step Test

To perform a step test (step operation) is to execute the program line by line. After one line of the program is executed, the program is halted. After executing a logic instruction, the next line becomes the current line and the cursor moves to the next line, but for the motion instruction, the cursor stays at the line that execution is completed.

Specifying the step mode (single step)

To specify the step mode, press the STEP key on the teach pendant. When the step mode is specified, the STEP LED on the teach pendant is lit. The STEP LED is off when continuous operation is specified.

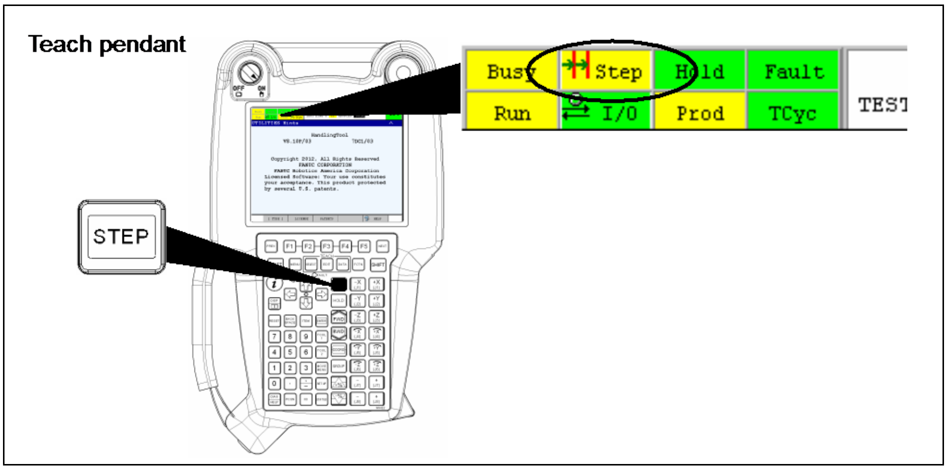


Fig. 6.3.2 (a) STEP key

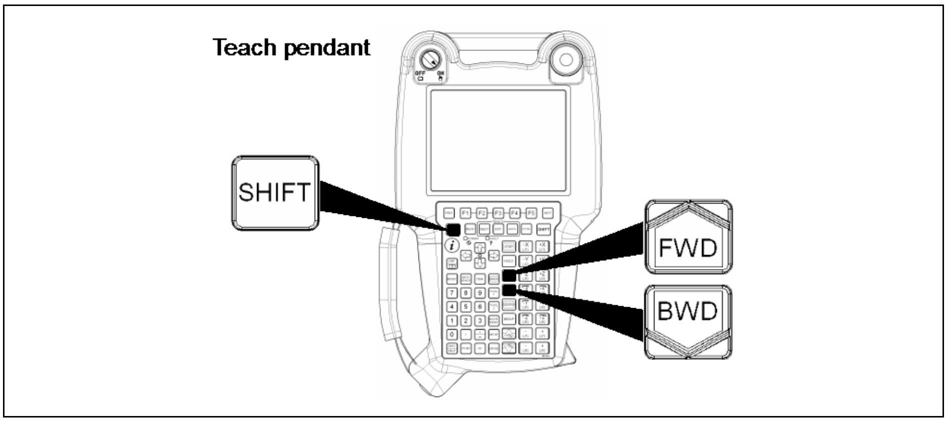


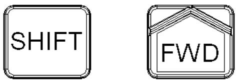
Fig. 6.3.2 (b) Starting step operation

Step operation can be performed in two ways: Forward execution and backward execution.

Forward execution

NOTE
In case that the following option is installed, the procedure of this operation differs.
- J591 Robot operation without shift key function (Refer to the section 33.2 Test execution operation without shift in the FANUC Robot series Optional Function OPERATOR'S MANUAL (B-83284EN-2).)

In forward execution, the program is executed in normal order. To perform forward execution of the program, press and hold down [SHIFT] key, then press and release [FWD] key on the teach pendant.

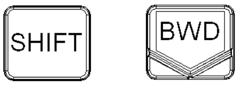


When a program is started, the program is executed for one line pointed to by the cursor, then the program is halted.
When a motion instruction is executed, the cursor is held at the executed line. When a logic instruction is executed, the cursor is moved to the next line.
Each time forward execution 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

> **NOTE**
> In case that the following option is installed, the procedure of this operation differs.
> - J591 Robot operation without shit key function (Refer to the section 33.2 Test execution operation without shift in the FANUC Robot series Optional Function OPERATOR'S MANUAL (B-83284EN-2).)

In backward execution, the program is executed in reverse order. To perform backward execution of the program, press and hold down [SHIFT] key, then press and release [BWD] key on the teach pendant.



- During backward execution, only the motion instructions can be executed. However, a skip instruction, time before instruction, time after instruction, soft float instruction, and other optional move instructions are ignored while the program is executed. After one line of the program is executed, the cursor is moved to the previous line.
- The instruction before the line where the following program instructions is taught can not be executed in backward execution. When you execute these instructions in backward execution, the cursor moves to the line following the line that contains taught these instructions:
  - Halt instruction (PAUSE)
  - Abort instruction (ABORT)
  - Program end instruction (END)
  - Jump instruction (JMP LBL[ ])
  - User alarm instruction (UALM[ ])
  - Execution instruction (RUN)
- The following program instructions cannot be executed:
  - Incremental instruction (INC)
- A blank line does not affect the execution of the program. (Both Forward and Backward execution)

When the terminated program is restarted, the motion instruction in the line pointed to by the cursor is executed, then the program is halted.

Each time backward execution of the program is started, the program is executed using the motion format and feed rate 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 (Start point of an arc motion in normal program execution) along the path which passes through the passing point 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 feed rate specified in the current line.

To disable backward execution of the program while the program is being executed, insert the halt instruction (PAUSE) into the desired location. After the halt instruction is executed, the cursor returns to the position it was at before the program was executed.
When the halt instruction is specified in the line before the line at the cursor is pointed at, backward execution of the program is disabled. To restart backward execution of the program, move the cursor to the line before the line that contains the halt instruction (two lines before the line at which the cursor is pointed).

Inter-program reverse program execution

With the inter-program reverse operation function, control can be returned from a subprogram to the main program that called the subprogram by performing reverse operation (SHIFT + BWD).

NOTE
1. Even if a subprogram exists during reverse operation of a main program, the subprogram cannot be called.
2. When program termination occurs within a subprogram, control cannot be returned to the main program.

When reverse execution is performed from a subprogram to the main program, the cursor stops at the line of the instruction that calls the subprogram taught in the main program.

Sample program

Example: When reverse operation is performed starting from the fourth line of a subprogram

```
Main_Prg
1:
2:R[1]=R[1]+1
3:J P[1] 100% FINE
4:IF R[1]=100, JMP LBL[100]
5:CALL Sub_Prog
6:
       .
       .
[End]

Sub_Prog
1:DO[1]=ON
2:DO[2]=ON
3:L P[2] 1000mm/sec FINE
4:L P[3] 1000mm/sec FINE
[End]
```

1. Start reverse operation with the cursor positioned to the fourth line of the subprogram.
2. Reverse operation (SHIFT + BWD) from P[3] to P[2]. The cursor is positioned to the third line of the subprogram.
3. Reverse operation (SHIFT + BWD) to the fifth line of the main program (CALL SUBPROGRAM). The cursor is positioned to the fifth line of the main program.
4. Reverse operation (SHIFT + BWD) from P[2] to P[1]. The cursor moves from the fifth line to third line of the main program.

Program end in backward execution

If the system variable $BWD_ABORT is set to TRUE, when the first line of the program is finished to be executed during the backward execution, this program ends.

Procedure 6-7 Step test

Condition

- The teach pendant must be enabled.
- The single-step mode must be set.
- The system must be in the operation enable state.
- No one must be in the operating area. No obstacle must be placed in the operating area.

> **NOTE**
> In case that the following option is installed, the procedure of this operation differs.
> - J591 Robot operation without shift key function (Refer to Procedure 33-2 Step test(in case robot operation without [SHIFT] key function is installed) in FANUC Robot series Optional Function OPERATOR'S MANUAL (B-83284EN-2).)

Step

1. Press [SELECT] key. The program selection screen will be displayed.
2. Select the program to be tested and press [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 safety fence is sound. Otherwise, injury or property damage would occur. If the program needs to be stopped before it terminates, the operator should release [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 [SHIFT] key, then press and release [FWD] key. Do not release [SHIFT] key until execution of the program is completed.
   - To perform backward execution of the program, press and hold down [SHIFT] key, then press and release [BWD] key. Do not release [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 [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 [SHIFT] key, then press and release [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 test 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.

> **NOTE**
> In case that the following option is installed, the procedure of this operation differs.
> - J591 Robot operation without shift key function (Refer to Procedure 33-3 Continuous test (using the teach pendant) (in case robot operation without [SHIFT] key function is installed) in the FANUC Robot series Optional Function OPERATOR’S MANUAL (B-83284EN-2).)

Step

1. Press [SELECT] key. The program selection screen is displayed.
2. Select the program to be tested and press [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 safety fence is sound. Otherwise, injury or property damage would occur. If the program needs to be stopped before it terminates, the operator should release [SHIFT] key or deadman switch or press the HOLD or emergency stop button.

6. Press and hold down [SHIFT] key, then press and release [FWD] key. Hold down [SHIFT] key until the execution of the program is completed. When [SHIFT] key is released, the program is halted.

   The program is executed to the end, then forcibly terminated. The cursor is returned to the first line of the program.

Procedure 6-9 Continuous test operation (started from the operator’s panel)

Condition

- The operator’s panel must be in the enabled state.
- Continuous operation mode must be set. (The step lamp must not be lit.)
- The system must be ready for operation.
- Nobody must be within the work area. There must be no obstacles.

Step

1 Press [SELECT] key. The program list screen will be selected.
2 Select a program to be tested, and press [ENTER] key. The program edit screen will be displayed.
3 Set continuous operation mode. Check that the step lamp is not lit. (If the STEP lamp is on, press the STEP key to turn it off.)
4 Position the cursor to the first line.
5 Place the system in local mode. (For how to switch to local mode, see the description of Remote/Local setting in Section 3.15,"SYSTEM CONFIG MENU".)

> ⚠ **WARNING**
> The execution of the program instructions starts in the next step. The execution causes the robot to make a motion, which may produce unpredictable results. The operator should check that no persons and no unnecessary equipment is in the work area and that each part of the safety fence is sound. Otherwise, injury or property damage would occur. If the program needs to be stopped before it terminates, the operator should release [SHIFT] key or deadman switch or press the HOLD or emergency stop button.

6 Press the start button on the operator's panel/box. Program execution is performed up to the end of the program then terminated forcibly. The cursor returns to the first line of the program.

6.3.4 Program Look/Monitor

When the program is executed, the screen of the teach pendant becomes a monitor screen by which the execution of the program is displayed. In the monitor screen, the cursor moves to follow the line which is executed and you can not edit a program.

Program monitor screen

```
PROGRAM1  LINE 1  T1  RUNNING  JOINT        30%
PROGRAM1
                                        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
      7: J  P[7] 100% FINE
      8: J  P[8] 100% FINE
      9: J  P[9] 100% FINE
   [End]

          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  T1  RUNNING  JOINT     30%
PROGRAM1
                                                  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
      7: J  P[7] 100% FINE
      8: J  P[8] 100% FINE
      9: J  P[9] 100% FINE
   [End]
    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 0  T1  ABORTED  JOINT     30%
PROGRAM1
                                                  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
      7: J  P[7] 100% FINE
      8: J  P[8] 100% FINE
      9: J  P[9] 100% FINE
   [End]

  POINT                              TOUCHUP   >
```

6.4 MANUAL I/O CONTROL

Under manual I/O control, signals are transmitted between the robot and peripherals before the program is executed.
The manual I/O control refers to the following items:

- Forced output
- Simulated output and simulated input
- Wait release

6.4.1 Forced Output

Forced output is to manually turn digital output signals on or off. For the group output and the analog output, specify the value.

Procedure 6-10 Forced output

Condition

- Assignment of the signals to be output must be completed.

Step

1 Press [MENU] 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 will be displayed.
If the input screen is displayed, press the F3, IN/OUT key to change the input screen to the output screen.

```
I/O Digital Out
           #   SIM STATUS              1/512
      DO[   1]  U  OFF  [                ]
      DO[   2]  U  OFF  [                ]
      DO[   3]  U  OFF  [                ]
      DO[   4]  U   ON  [                ]
      DO[   5]  U   ON  [                ]
      DO[   6]  U  OFF  [                ]
      DO[   7]  U  OFF  [                ]
      DO[   8]  U   ON  [                ]
      DO[   9]  U  OFF  [                ]
      DO[  10]  U  OFF  [                ]
      DO[  11]  U  OFF  [                ]
    Sorted by port number.

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

> ⚠ **WARNING**
> Forced output activates connected equipment. Before executing the forced output, the operator should check which equipment is connected to the digital output and what operation the forced output would cause. Otherwise, injury or property damage could occur.

5 Move the cursor to the status field for the signal number to be changed, then press the F4, ON or F5, OFF key to change the signal output setting.

```
I/O Digital Out
         #   SIM STATUS              1/512
      DO[   1]  U    ON  [              ]
      DO[   2]  U   OFF  [              ]
      DO[   3]  U   OFF  [              ]
      DO[   4]  U    ON  [              ]
      DO[   5]  U    ON  [              ]
      DO[   6]  U   OFF  [              ]
      DO[   7]  U   OFF  [              ]
      DO[   8]  U    ON  [              ]
      DO[   9]  U   OFF  [              ]
      DO[  10]  U   OFF  [              ]
      DO[  11]  U   OFF  [              ]
    Sorted by port number.

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

Manual forced group output

6 Press F1, [TYPE]. The screen change menu will be displayed.
7 Select Group. The group output screen will be displayed.

```
I/O Group Out
         #   SIM VALUE               1/100
      GO[   1]  U      1 [              ]
      GO[   2]  U     10 [              ]
      GO[   3]  U     23 [              ]
      GO[   4]  *    *   [              ]
      GO[   5]  *    *   [              ]
      GO[   6]  *    *   [              ]
      GO[   7]  *    *   [              ]
      GO[   8]  *    *   [              ]
      GO[   9]  *    *   [              ]
      GO[  10]  *    *   [              ]
      GO[  11]  *    *   [              ]
    Sorted by port number.

  [ TYPE ]  CONFIG  IN/OUT                >
```

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
         #   SIM VALUE               1/100
      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, group or Robot I/O communicate with peripherals. This function is used to execute the program or to test the I/O instruction when connection of I/O with peripherals is not completed. Simulated input/output can be used for digital, analog, group and Robot I/O.
To enable simulated input/output, set the simulated flag, S.

Simulated output

The simulated output function internally changes the signal state using the I/O instruction of the program or manual output, but does not change the state of output to peripherals. This function holds the state of output to peripherals when the simulated flag is set.
When the simulated flag is reset, the output is restored to the original state.

Simulated input

The simulated input function internally changes the signal state with the I/O instruction of the program or manual input. The state of input from peripherals is ignored, and the signal state is not changed internally. When the simulated flag is reset, the input enters the current state.
Refer to 6.3.1,"Specifying test execution" to specify whether I/O signal is disabled in the test execution.

Procedure 6-11 Simulated input / output

Condition

- The input/output signal has been allocated.

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select I/O. The I/O screen will be displayed.
3 Press F1, [TYPE]. The screen change menu will be displayed.
4 Select Digital. Digital I/O screen will be displayed.

```
I/O Digital In
          #   SIM STATUS              1/512
     DI[   1]  U   OFF  [Digital signal 1 ]
     DI[   2]  U   OFF  [Digital signal 2 ]
     DI[   3]  U   OFF  [Digital signal 3 ]
     DI[   4]  U    ON  [Digital signal 4 ]
     DI[   5]  U    ON  [Digital signal 5 ]
     DI[   6]  U   OFF  [Digital signal 6 ]
     DI[   7]  U   OFF  [Digital signal 7 ]
     DI[   8]  U    ON  [Digital signal 8 ]
     DI[   9]  U    ON  [Digital signal 9 ]
     DI[  10]  U   OFF  [Digital signal 10]
     DI[  11]  U   OFF  [Digital signal 11]
   Sorted by port number.

 [ 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, SIMULATE or F5, UNSIM key to change the simulated setting.

```
I/O Digital In

     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    ON  [Digital signal 1 ]

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

6.4.3 Wait Release

When a WAIT instruction in a program waits until the I/O conditions are satisfied, the wait release function skips this instruction, and halts program execution at the next line. Wait release is enabled only when a program is being executed. Wait release is performed by choosing from the function menu.

Procedure 6-12 Wait release

Condition

- Program execution is currently in the I/O wait state.

```
SAMPLE3
                                   11/20
      10: J  P[5] 100% FINE
      11:    WAIT RI[1]=ON
      12:    RO[1]=ON
```

Step

1 Press [FCTN] key to display the function menu.
2 Select 7 RELEASE WAIT.
The I/O wait is skipped, and the cursor moves to the next line. The program will be then halted. When program execution is restarted, the next instruction is executed.

6.5 OPERATING THE HAND MANUALLY

To operate the hand manually using the teach pendant, hand instruction must be assigned to the manual operation screen when macro instructions are set.
When teaching the operation of the hand, check whether the hand can actually hold a workpiece at the target position by operating the hand.

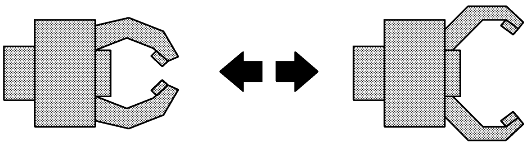


Fig. 6.5 Operating the hand manually

Procedure 6-13 Operating the hand manually

Condition

- The teach pendant must be enabled.
- No one must be in the operating area. All obstacles must be removed from the operating area.
- The hand instruction must be defined as a macro for manual operation (MF).

Step

1 Press and hold down the deadman switch, then turn on the teach pendant enable switch.
  - Press the TOOL1 (or TOOL2) key. The Tool 1 (or Tool 2) screen will be displayed.

```
Tool 1
                                    1/3
       Instruction
     1 Open hand 1
     2 Close hand 1
     3 Relax hand 1




 Press SHIFT-EXEC(F3) to run program
                EXEC
```

- Otherwise, select MANUAL FCTNS from the screen menu to display the manual operation screen.

```
MANUAL Macros
                                              1/3
        Instruction
      1 Open hand 1
      2 Close hand 1
      3 Relax hand 1



    Press SHIFT-EXEC(F3) to run program

   [ TYPE ]           EXEC
```

2 Move the cursor to the desired macro instruction. Press and hold down [SHIFT] key, then press the F3, EXEC.

6.6 AUTOMATIC OPERATION

Peripheral I/O can be used to automatically start a program and operate a production line. (See Section 3.8, "SETTING AUTOMATIC OPERATION".)

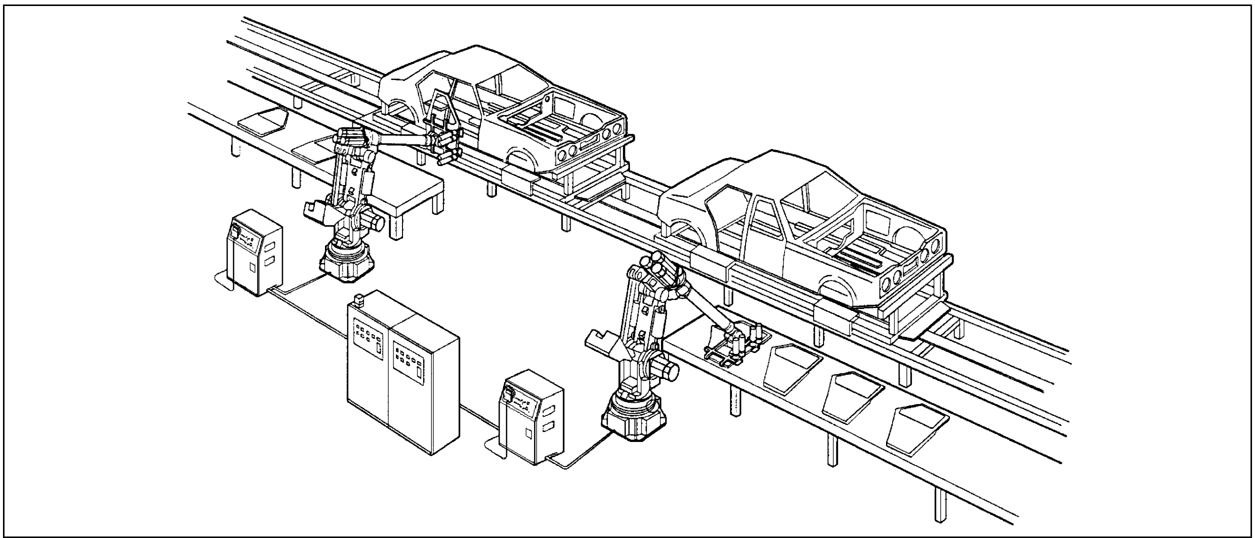


Fig. 6.6 Automatic operation of robot system

6.6.1 Automatic Operation by Robot Start Request (RSR)

The robot start request (RSR) function allows a remote device to select and start a program through the peripheral device I/O. This function uses eight robot start request signals (RSR1 to RSR8).

Procedure 6-14 Automatic operation by robot start request (RSR)

Condition

- RSR settings are completed. (See Subsection 3.8.1.)
- Remote mode is set.
- The system is ready for operation.
- Nobody must be within the work area. There must be no obstacles.

> ⚠ **WARNING**
> Applying this procedure starts automatic operation which causes the robot to move. An unpredictable operation could occur. Check to ensure that nobody is in the work area, that there are no unnecessary objects in the work space, and that the safety fence is normal. Also, check that all the automatic operation conditions are set correctly. Otherwise, personal injury or damage to the facilities could occur.

Step

1. Set the enable switch on the teach pendant to OFF.
2. Place the system in remote mode. (For how to switch to remote mode, see the description of Remote/Local setup in Section 3.15, "SYSTEM CONFIG MENU".)
3. Send the robot start signal (RSR1 to RSR8 input) of a target RSR number to the controller. The RSR program is placed in a queue.
4. To stop the program currently being executed, use the emergency stop button or hold button, or the immediate stop (*IMSTP input), hold (*HOLD input), or cycle stop (CSTOPI input) signal.
5. To cancel a job in the queue, use the cycle stop signal (CSTOPI input).
6. To restart a halted program, use the external start signal (START input).

6.6.2 Automatic Operation with Program Number Selection (PNS)

The program number selection (PNS) function enables selection or checking of a program, using the peripheral I/O, from the remote controller. Eight input signals, PNS1 to PNS8, specify a PNS program number.

Procedure 6-15 Automatic operation by program number selection

Condition

- PNS setting must be completed (See Subsection 3.8.2).
- The remote condition must be satisfied.
- The operation enable condition must be satisfied.
- No one must be within the work area. There must be no obstacles.

> ⚠ **WARNING**
> Start automatic operation as follows: When the robot starts operation, an unexpected situation may occur. To prevent any problem from occurring, be sure to check that no one is in the work area, that the work area is free from unnecessary equipment, that the safety barrier is in place, and that all the automatic operation conditions are correctly specified. Otherwise, the robot may injure a person or damage the equipment in the work area.

Step

1. Turn off the teach pendant enable switch.
2. Place the system in remote mode. (For how to switch to remote mode, see the description of Remote/Local setup in Section 3.15, "SYSTEM CONFIG MENU".)
3. Send the program number selection signals (PNS1 to PNS8 inputs) indicating a target PNS number and the PNS strobe signal (PNSTROBE input) to the controller. A PNS program is then selected.
   The controller outputs the selected program number signals (SNO1 to SNO8 inputs) and PNS acknowledge signal (SNACK output) for confirmation.
4. Send an external start signal (PROD_START input). The selected program is then started.
5. To stop the program currently being executed, use the emergency stop button or hold button, or the immediate stop (*IMSTP input), hold (*HOLD input), or cycle stop (CSTOPI input) signal.
6. To restart a halted program, use the external start signal (START input).

6.6.3 External Override Selection Function

The external override selection function changes feed rate override by turning on or off digital input (DI) signals.
Two DI signals are defined. These two signals can be combined in four different ways. So four types of feed rate override can be selected.

```
OVERRIDE SELECT
                                          1/7
      1  Function Enable   :ENABLE

      2 Signal 1           : DI[   1][OFF]
      3 Signal 2           : DI[  32][ON ]

        Signal 1     Signal 2      Override
      4    OFF          OFF           15 %
      5    OFF          ON            30 %
      6    ON           OFF           65 %
      7    ON           ON           100 %

   [ TYPE ]                 ENABLE   DISABLE
```

To enable the external override selection function, the following requirements must be satisfied:

- The external override selection function must be enabled. (OVERRIDE SELECT on the setting screen)
- The remote mode must be set.

When the external override selection function is enabled, the following occurs:

- The override key of the teach pendant is practically disabled. (The changed value is quickly returned to the setting value by the external override selection.)
- The override instruction has no effect to the override value.
- You can not change the settings of DI signal number and Override. Before these settings can be modified, Function Enable : DISABLE must be set.
- When this function is effective at turning off the power of the controller, the override will get the value which had been set by this function when turning on it again.
- It is possible to specify the same number as two DI signal numbers. In this case, only the combination of ON-ON or OFF-OFF has the meaning.

Moreover, note the following:

- After this function is disabled because the remote condition is not satisfied, the override keeps to remain the value specified by this function in effective until the value is changed by the teach pendant or override instruction.

Set this function on the external override selection setting screen (6 SETUP Ovrd Select).

Procedure 6-16 Selecting an external override

Step

1 Press [MENU] key 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
                                          1/7
      1  Function Enable   :DISABLE

      2 Signal 1           : DI[****][***]
      3 Signal 2           : DI[****][***]

        Signal 1     Signal 2      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 DI signals.

```
OVERRIDE SELECT

      2 Signal 1           : DI[  11][***]
      3 Signal 2           : DI[****][***]

   [ TYPE ]
```

The states of DI signals are indicated. When *** is displayed, the setting of the function cannot be changed.

   c Feed rate override to be changed by turning on or off the signals

```
OVERRIDE SELECT
                                          4/7
      1  Function Enable   :ENABLE

      2 Signal 1           : DI[  11][ON ]
      3 Signal 2           : DI[  12][OFF]

        Signal 1     Signal 2      Override
      4    OFF          OFF            15 %
      5    OFF          ON             30 %
      6    ON           OFF            65 %
      7    ON           ON            100 %

   [ TYPE ]
```

6.7 ONLINE POSITION MODIFICATION

Online position modification replaces all the position data and move speeds in the move instructions within a certain range in a program at one time, according to the position modification condition, during program execution.

The following program information can be modified:

- Position data (position compensation)
- Move speed

Position data is modified by adding a position compensation value. A movement speed is modified by rewriting it. Up to ten positions modification conditions can be defined.

Position compensation value

A position compensation value is the difference between the current position and the correct position. The position data coded in the move instructions within a specified range of a program is rewritten by adding a position compensation value to the data.
If the position data resulting from modification falls outside the allowable axial movement range, an alarm is generated when the program is executed.

> ⚠ **CAUTION**
> If position compensation is performed during execution, it may take a while for the compensation to be reflected in actual operation.

The specifiable ranges (+/-) for the position compensation values are set in system variables $PRGADJ.$X_LIMIT to $R_LIMIT. The standard value is +/-26 mm for (X, Y, Z) and +/-0.5 degrees for (W, P, R). Any position compensation value falling outside these ranges cannot be set.

Move speed

Move speeds in the move instructions within a specified range of a program are replaced with specified speeds. The move speed for axial movement is replaced by the value specified in Joint speed, while the move speed for linear, circular and circle arc 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
    Program Adjust Schedules          1/10
      #  Program      Lines    Status
      1  SAMPLE1     22 - 29    ENABLED
      2  SAMPLE1     39 - 49    ENABLED
      3  SAMPLE3     10 - 14    DISABLED
      4  SAMPLE4    123 - 456   EDIT
      5  ********     1 - 0     EDIT
      6  ********     1 - 0     EDIT
      7  ********     1 - 0     EDIT
      8  ********     1 - 0     EDIT
      9  ********     1 - 0     EDIT
     10  ********     1 - 0     EDIT

   [TYPE]   DETAIL                      >
   COPY     CLEAR    CLR_ALL            >
```

Position modification condition detail screen

```
UTILITIES Prog Adj
                                          1/14
     Current schedule:  5 Status:EDIT
        Program name:
        1 SAMPLE2
        2 Starting line number:  1
        3 Ending line number:    30
        4 Offset relative to:    USER
        5 X adjustment:          5.000   mm
        6 Y adjustment:          0.000   mm
        7 Z adjustment:          -2.500  mm
        8 W adjustment:          0.000   deg
        9 P adjustment:          0.000   deg
       10 R adjustment:          0.000   deg
       11 Linear/Circular speed: 2000    mm/s
       12 Joint speed:           10      %
       13 Motion group:          ALL
          Group:  [1]
       14 Adjust [Y] for:              ROBOT

   [TYPE]   UNITS    SCHED    [CHOICE]          >
   COPY     CLEAR    CLR_ALL  [CHOICE]          >
```

Table 6.7(a) Online position modification settings

Item	Description
Program	Specifies the name of the target program for position modification.
Range	Specifies the range (the start and end lines) of the program lines to which position modifications are to be applied. **NOTE** The end line number must be greater than or equal to the start line number specified in item 2. When only one line is to be modified, the end line number must equal the start line number.
Offset relative to	User Modification is performed in reference to the user coordinate system. Tool Modification is performed in reference to the tool coordinate system.
Status	The position modification status indicates whether a specified position modification condition is reflected in the program. • EDIT : The position modification condition is being edited. • ENABLED : The position modification condition is reflected in the program. • DISABLED : The position modification condition is not reflected in the program.
X to R adjustment	Compensation values X to R indicate the position compensation amounts. Values (X, Y, Z) are in mm or inches, while values (W, P, R) are in degrees. The values specified here are included in the position data.
Motion speed **Joint speed**	These speed items replace the move speeds. Motion speed replaces the linear and circular and circle arc movement speed with a specified speed. Joint speed replaces the axial movement speed with a specified speed. ⚠ **CAUTION** Once the move speed is rewritten, the original speed cannot be restored.
Motion group	Select an operation group to be subjected to modification.
Adjust Y for	This item is displayed only when an integrated additional rail axis is set up as the seventh axis in group 1. The direction of the integrated additional axis is indicated in motion group. Specify the compensation target for the indicated direction. Robot: Modify only the position of the robot. Additional axis: Modify the position of the additional axis. All: Modify both the positions of the robot and the additional axis. If offset relative to is set to “Tool,” only the robot can be selected.

```
| [ TYPE ] | UNITS | SCHED | ENABLE |        | > |
   COPY      CLEAR   CLR_ALL                    >
```

Table 6.7(b) Function key for online position modification

Function key label	Description
UNITS	The position modification unit function key changes the units of the position modification values (mm or inches).
SCHED	The schedule function key 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 key copies a selected position modification condition into another condition number. After copying, EDIT is indicated as the modification status.
CLEAR	The position modification condition erase function key 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 [MENU] 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 will be displayed.

Position modification condition list screen

```
UTILITIES Prog Adj
    Program Adjust Schedules             1/10
       #  Program      Lines      Status
       1  SAMPLE1     22 - 29      ENABLED
       2  SAMPLE1     39 - 49      ENABLED
       3  SAMPLE3     10 - 14      DISABLED
       4  SAMPLE4    123 - 456     EDIT
       5  ********     1 - 0       EDIT
       6  ********     1 - 0       EDIT
       7  ********     1 - 0       EDIT
       8  ********     1 - 0       EDIT
       9  ********     1 - 0       EDIT
      10  ********     1 - 0       EDIT

| [ TYPE ] | DETAIL |        |        |        | > |
```

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.

Position modification condition detail screen

```
UTILITIES Prog Adj
                                            1/14
     Current schedule:  5 Status:EDIT
         Program name:
       1 SAMPLE2
       2 Starting line number:  0
       3 Ending line number:    0
       4 Offset relative to:    USER
       5 X adjustment:          0.000   mm
       6 Y adjustment:          0.000   mm
       7 Z adjustment:          0.000   mm
       8 W adjustment:          0.000   deg
       9 P adjustment:          0.000   deg
      10 R adjustment:          0.000   deg
      11 Linear/Circular speed: 0       mm/s
      12 Joint speed:           0       %
      13 Motion group:          ALL
         Group:  [1]
      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
                                             1/14
     Current schedule:  5 Status:EDIT
         Program name:
       1 SAMPLE2
       2 Starting line number:  1
       3 Ending line number:    30
       4 Offset relative to:    USER
       5 X adjustment:          5.000   mm
       6 Y adjustment:          0.000   mm
       7 Z adjustment:          -2.500  mm
       8 W adjustment:          0.000   deg
       9 P adjustment:          0.000   deg
      10 R adjustment:          0.000   deg
      11 Linear/Circular speed: 2000    mm/s
      12 Joint speed:           10      %
      13 Motion group:          ALL
         Group:  [1]
      14 Adjust [Y] for:              ROBOT

   [ TYPE ]  UNITS   SCHED  [CHOICE]          >
```

8 After completing the modification condition settings, press F4, ENABLE to reflect the position modifications in the target program. The result of ENABLE is reflected immediately if the program is being executed.

NOTE

1 To modify a position modification condition after making it valid, cancel the condition once, then modify it.
2 When move instructions include a position register or incremental instruction, modifications are not reflected.

9 To cancel a set modification condition, press F5, DISABLE. When DISABLE is used, the current position modification condition must be valid.

⚠ 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.
11 Press [PREV] key to redisplay the position modification list screen.

```
UTILITIES Prog Adj
    Program Adjust Schedules            1/10
       #  Program      Lines      Status
       1  SAMPLE1     22 - 29      ENABLED
       2  SAMPLE1     39 - 49      ENABLED
       3  SAMPLE3     10 - 14      DISABLED
       4  SAMPLE4    123 - 456     EDIT
       5  ********     1 - 0       EDIT
       6  ********     1 - 0       EDIT
       7  ********     1 - 0       EDIT
       8  ********     1 - 0       EDIT
       9  ********     1 - 0       EDIT
      10  ********     1 - 0       EDIT

   [ TYPE ]  DETAIL                      >
```

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.
13 To erase the set modification condition, press F2, CLR_ADJ on the next page.

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 two LEDs and the software LED in the status window on the teach pendant indicate the following statuses:

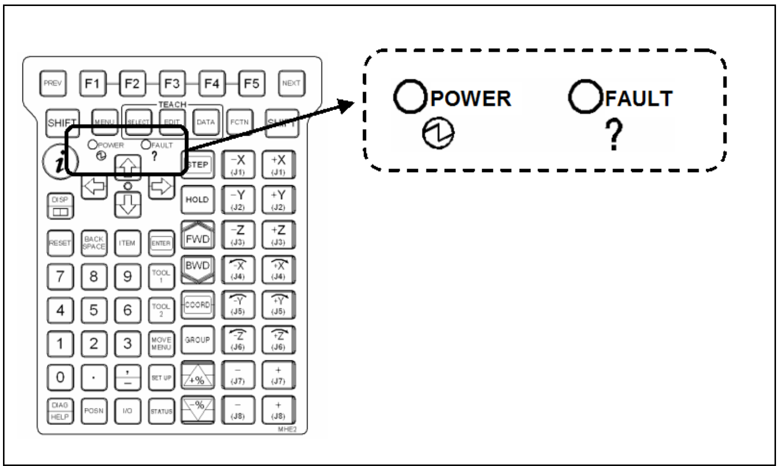


Fig. 7.1 (a) LEDs on the teach pendant

Table 7.1 (a) LEDs on the teach pendant

LED	Description
POWER	This LED indicates that the power of the controller is on.
FAULT	This LED indicates that an alarm has been issued.

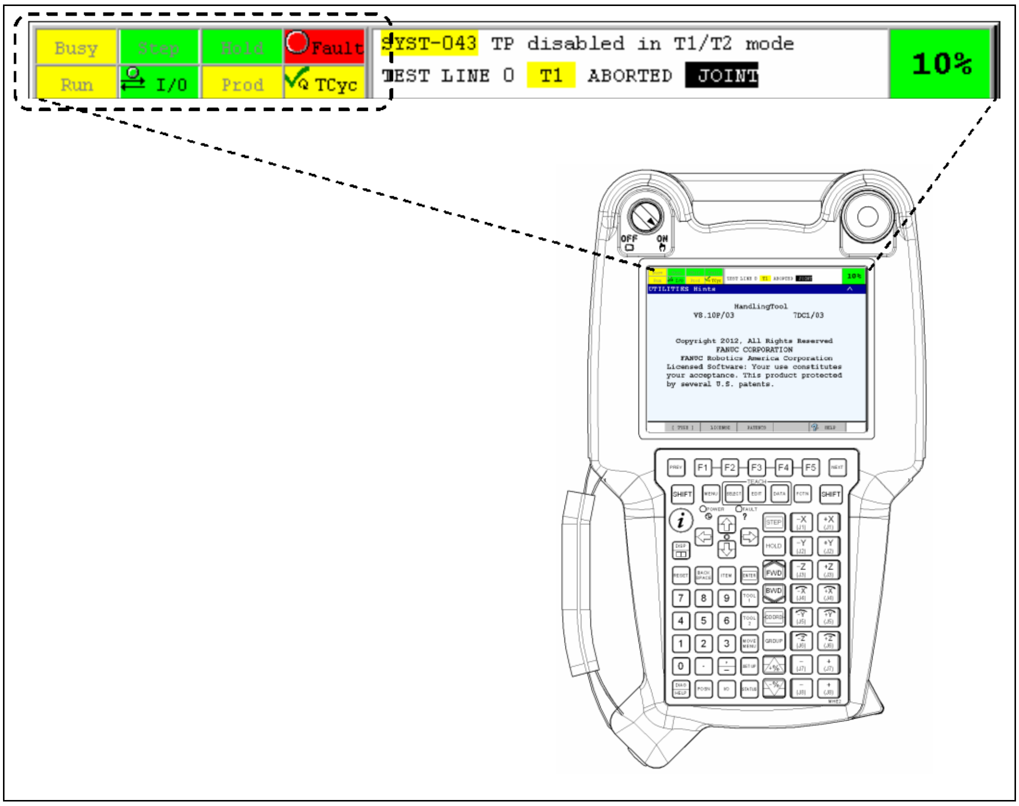


Fig. 7.1 (b) Software LEDs on the teach pendant

Table 7.1 (b) Software LEDs on the teach pendant

LEDs (Upper: On, Lower: Off)	Description
Busy	Indicates that the robot is working.
Step	Indicates that the robot is in the step operation mode.
Hold	Indicates that the HOLD button is being held or the HOLD signal is input.
Fault	Indicates that an alarm occurs.
Run	Indicates that a program is being executed.
I/O	Application-specific LED. This is a sample LED for a handling tool.
Prod	Application-specific LED. This is a sample LED for a handling tool.
TCyc	Application-specific LED. This is a sample LED for a handling tool.

7.2 USER SCREEN

A message instruction for the program being executed is displayed on this screen. (See Subsection 4.15.7.) When a message instruction is executed, the screen display automatically switches to the user screen.

Procedure 7-1 User screen display

Step

1. Press [MENU] key.
2. Select “9 USER”.

> **NOTE**
> - When a message instruction is not executed, nothing is displayed on this screen.
> - 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 [MENU] key to display the screen menu.
2. Press ”NEXT”, then select “DATA”.
   Alternatively, instead of steps 1 and 2 above, the user can press [DATA] key.
3. Press F1, [TYPE].
4. Select Registers. The register screen will be displayed.

```
DATA Registers
                                       1/200
    R[   1:                      ]=0
    R[   2:                      ]=0
    R[   3:                      ]=0
    R[   4:                      ]=0
    R[   5:                      ]=0
    R[   6:                      ]=0
    R[   7:                      ]=0
    R[   8:                      ]=0
    R[   9:                      ]=0
    R[  10:                      ]=0
    R[  11:                      ]=0
    Enter value

   [ TYPE ]
```

> **⚠ WARNING**
> Registers are used in a program. Never change the value of a register before checking how the register is used in the system. Otherwise, the program can be adversely affected.

5. To enter a comment, use the following procedure:
   a. Move the cursor to a desired register number field, then press the [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.

```
DATA Registers
                                         1/200
    R[  1:                    ]=12
    R[  2:                    ]=0
    R[  3:                    ]=0
    R[  4:                    ]=0
    R[  5:                    ]=0
    R[  6:                    ]=0
    R[  7:                    ]=0
    R[  8:                    ]=0
    R[  9:                    ]=0
    R[ 10:                    ]=0
    R[ 11:                    ]=0
    Enter value

   [ TYPE ]
```

Programming example

7 Registers are used in programs when the following are specified:

- Register instruction (See Subsection 4.5.1)
- Indirect specification of arguments (See Section 4.2)

```
SAMPLE4
                                          1/8
       1:   R[1]=0
       2:   LBL[1]
       3:   CALL PROGRAM_A
       4:   R[1]=R[1]+1
       5:   IF R[1]=<10, JMP LBL[1]
       6:   CALL PROGRAM_B
       7:   END
     [End]

   [ INST ]                            [EDCMD]  >
```

In this program, the program A is repeated 11 times, and the program B is executed, then the program execution terminates.

7.4 POSITION REGISTERS

A position register is a variable for holding position data. One hundreds position registers are provided. The position register screen is used to display and set registers.

Procedure 7-3 Position register setting

Step

1 Press [MENU] 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 [DATA] key.

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

4 Select Position Reg. The position register screen will be displayed.

```
DATA Position Reg
                                               1/100
    PR[  1:                  ]=*
    PR[  2:                  ]=*
    PR[  3:                  ]=*
    PR[  4:                  ]=*
    PR[  5:                  ]=*
    PR[  6:                  ]=*
    PR[  7:                  ]=*
    PR[  8:                  ]=*
    PR[  9:                  ]=*
    PR[ 10:                  ]=*
    PR[ 11:                  ]=*
    Press ENTER

   [ TYPE ]  MOVE_TO   RECORD   POSITION   CLEAR
```

⚠ WARNING

Position registers are used in a program. Never change the value of a position register before checking how the register is used in the system. Otherwise, the program can be adversely affected.

5 To enter a comment, use the following procedure:
   a Move the cursor to a desired position register number field, then press the [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 [SHIFT] key.

```
DATA Position Reg
                                               1/100
    PR[  1:REF POSITION      ]=R
    PR[  2:                  ]=*
    PR[  3:                  ]=*
    PR[  4:                  ]=*
    PR[  5:                  ]=*
    PR[  6:                  ]=*
    PR[  7:                  ]=*
    PR[  8:                  ]=*
    PR[  9:                  ]=*
    PR[ 10:                  ]=*
    PR[ 11:                  ]=*

   [ TYPE ]  MOVE_TO   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 [SHIFT] key.

```
DATA Position Reg
                                        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.

```
DATA Position Reg
                                        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 will be displayed. To change a value, move the cursor to the desired field, then enter a new value.

```
DATA Position Reg
     PR[1] UF:F  UT:F       CONF:NUT 000
     X   1500.374   mm  W     40.000  deg
     Y   -342.992   mm  P     10.000  deg
     Z    956.895   mm  R     20.000  deg
    Position Detail
    PR[  1:REF POSITION       ]=R

                    CONF     DONE   [REPRE]
```

10 To change the configuration, press F3, CONF. Move the cursor to a desired field, then change joint placement data using the ↓ and ↑ keys.

```
DATA Position Reg
     PR[1] UF:F  UT:F       CONF:NUT 000
     X   1500.374   mm  W     40.000  deg
     Y   -342.992   mm  P     10.000  deg
     Z    956.895   mm  R     20.000  deg
    Position Detail
    PR[  1:REF POSITION       ]=R

     Select Flip or Non-flip 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.

```
DATA Position Reg
     PR[1] UF:F  UT:F       CONF:NUT 000
     X   1500.374   mm  W     40.000  deg
     Y   -342.992   mm  P     10.000  deg
     Z    956.895   mm  R     20.000  deg
    Position Detail                REPRE 1
    PR[  1:REF POSITION       ]=R 1 Cartesian
     Select Flip or Non-flip by UP/D 2 Joint

                    POSITION  DONE   [REPRE]
```

```
DATA Position Reg
     PR[1] UF:F  UT:F
     J1      34.304 deg   J4    27.089 deg
     J2      56.008 deg   J5   -10.503 deg
     J3    -121.672 deg   J6     0.347 deg
    Position Detail
```

12 To change the display to the additional axes (subgroup), press F2, PAGE.

```
DATA Position Reg
      PR[1] UF:F  UT:F
      E1      0.204 deg
      E2    100.204 deg
      E3     -0.894 deg
     Position Detail
```

13 Upon completion of setting, press F4, DONE.

```
DATA Position Reg
                                       1/100
    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:                   ]=*
    PR[  7:                   ]=*
    PR[  8:                   ]=*
    PR[  9:                   ]=*
    PR[ 10:                   ]=*
    PR[ 11:                   ]=*

  [ TYPE ]  MOVE_TO  RECORD  POSITION  CLEAR
```

Program example

14 The position register can be used in the program as the following case:

- Position data of motion instruction (See Subsection 4.3.2)
- Position register instruction and offset instruction, etc. (See Section 4.5 and Subsection 4.3.5)

```
SAMPLE4
                                       12/18
     12:   LBL[1]
     13:   OFFSET CONDITION PR[1]
     14: L  PR[2] 1000mm/sec CNT100 Offset
     15:   PR[3,6]=R[10]
     16: L  PR[3] 1000mm/sec CNT100
     17: L  PR[4] 1000mm/sec CNT100 Offset
    [End]

  [ 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 [MENU] 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 [DATA] key.
3 Press F1, [TYPE].
4 Select “Pallet register”. The pallet register screen will be displayed.

```
Data:Pallet Registers
                                           1/32
    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]
    PL[ 11:               ]=[   1,    1,    1]
    Enter value

  [ TYPE ]
```

⚠ WARNING
Palletizing registers are used in a program. Never change the value of a palletizing register before checking how the register is used in the system. Otherwise, the program can be adversely affected.

5 To enter a comment, use the following procedure:
  a Place the cursor on the comment line, then press the [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.

```
Data:Pallet Registers
                                           1/32
    PL[  1:BOX PALLET     ]=[   2,    1,    1]
    PL[  2:               ]=[   1,    1,    1]
    PL[  3:               ]=[   1,    1,    1]
    PL[  4:               ]=[   1,    1,    1]
    PL[  5:               ]=[   1,    1,    1]
```

7.6 STRING REGISTERS

The DATA String Reg screen displays the current value of each string register in the system. You can change the value of any string register and add comments using the DATA String Reg Screen.

Procedure 7-5 Displaying a string register screen

Step

1 Press [MENU] 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 [DATA] key.
3 Press F1, [TYPE] to display the screen change menu.
4 Select String Reg. The string register screen will be displayed.

```
DATA String Registers
                                         1/25

    SR[ 1:                    ]=
    SR[ 2:                    ]=
    SR[ 3:                    ]=
    SR[ 4:                    ]=
    SR[ 5:                    ]=
    SR[ 6:                    ]=
    SR[ 7:                    ]=
    SR[ 8:                    ]=
    SR[ 9:                    ]=
    SR[10:                    ]=

   [ TYPE ]  DETAIL   IMPORT
```

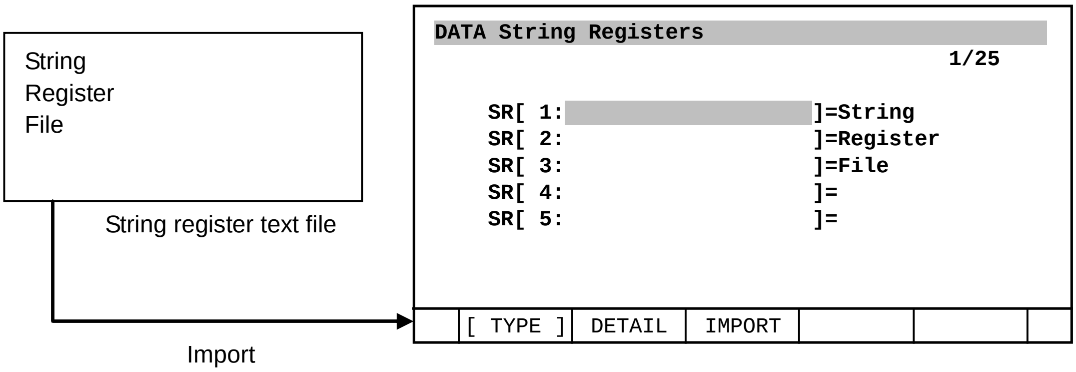
5 To add a comment
  a. Move the cursor to the string register number and press the [ENTER] key.
  b. Select the way of naming the comment.
  c. Press a desired function key, and then enter characters.
  d. Upon completion of input, press the [ENTER] key.
6 To change the value
  a. Move the cursor to the string register number on the right side of the = and press the [ENTER] key.
  b. Select the way of naming the value.
  c. Press a desired function key, and then enter characters.
  d. Upon completion of input, press the [ENTER] key.
7 To view string register values too long for the screen
  a. Move the cursor to the string register number.
  b. Press F2, DETAIL.
  c. Press PREV to return from the DETAIL screen.
8 To import a string register text file
  The file of string register is the text file. It is possible to load the value to the string register by importing the string register file.
  a. Press F3, IMPORT.
  b. Move the cursor to the file to be imported.
  c. Press the [ENTER] key.

NOTE
Importing a string register text file will overwrite all current comments and values stored in the string registers.

The string register import text file can have two formats.

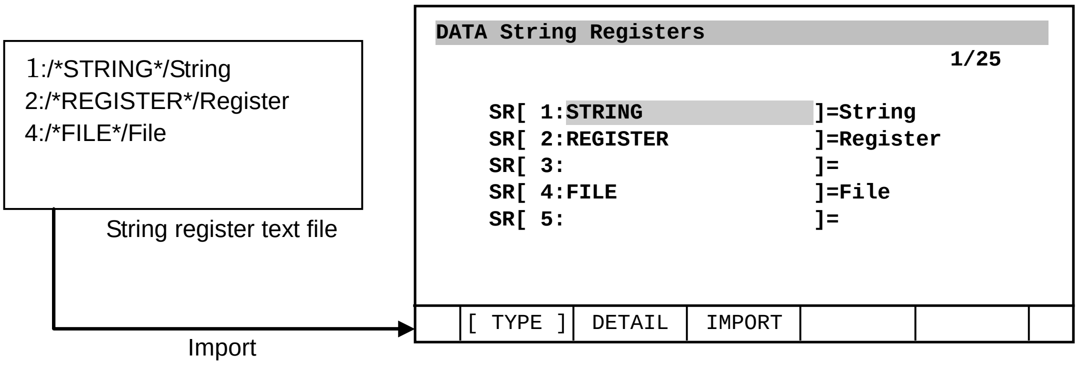
- The simple format containing only values for each string registers
  The values will be imported starting with string register number 1 and increment to the next string register number on the end of the line.

Example:



- The full format containing string register number, comment, and values
  A specific string register comment and value can be set by importing the string register text file written in the format such as “number: /*comment*/ value”.

Example:



7.7 4D GRAPHICS

7.7.1 4D GRAPHICS Display

In 4D graphics function, robots, tools, parts and various work cells can be displayed as 3D models.
And internal data such as positions taught in a program can be visualized as 4^{th} dimensional information.
4D means the fusion of 3D robot model and 1D internal data as 4^{th} dimension of information.
The robot model moves as the real robot moves. In machine lock, only the robot model can be moved and the direction of the movement of the robot can be previewed.

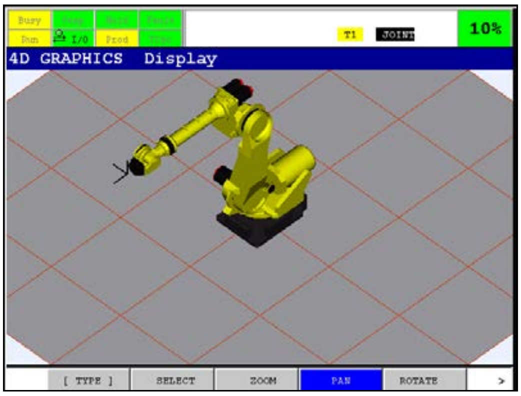
> ⚠ **WARNING**
> When moving the robot with the pendant enabled, be sure to watch the robot instead of watching the teach pendant screen. After the robot is in a safe state you can examine the pendant graphics.

> ⚠ **WARNING**
> 4D graphics might no be an accurate representation of the real world, so actual program verification with the robot arm is still required.

Procedure 7-6 Displaying 4D GRAPHICS screen

Step

1 Press [MENU] key to display the screen menu.
2 Press “0 NEXT”, then select “5 4D GRAPHICS”.
3 Alternatively, instead of steps 1 and 2 above, the user can press ***i*** key and POSN key at the same time.



7.7.1.1 Graphic models

4D GRAPHICS screen displays the following graphic models.
- Robot model
- Tool center point
- Cell floor

Robot model
A graphic model of ordered robot is displayed.

Tool center point
A symbol of Tool center point is displayed.

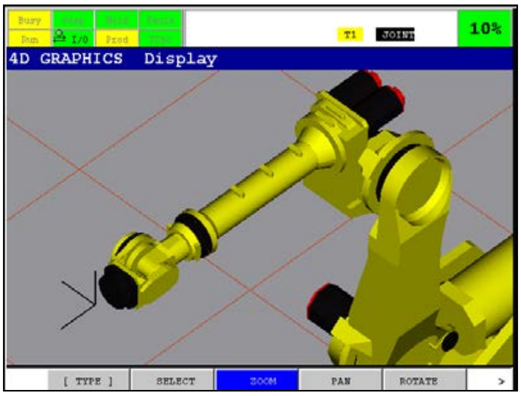
Cell floor
A reference floor with a 1 meter grid. The position of cell floor can be set in SETUP Frames screen.

7.7.1.2 Operation procedure

In 4D GRAPHICS screen, view is adjusted by the following operation modes.
- ZOOM
- PAN
- ROTATE

And the following functions are prepared.
- Preset Views
- User Views

ZOOM



ZOOM consists of changing the magnification. Increasing the magnification makes the objects larger but the field of view is narrow. Press F3, ZOOM to set the system to zoom mode. Then, the label F3, ZOOM turns blue in color.

There are two ways for ZOOM operation. One is the *i*Pendant key input and the other is touch panel operation.

Zoom by *i*Pendant key input

Increase the magnification

- Press the up arrow key on *i*Pendant.
- Press the SHIFT + up arrow key on *i*Pendant. (High magnification)

Decrease the magnification

- Press the down arrow key on *i*Pendant.
- Press the SHIFT + down arrow key on *i*Pendant. (High magnification)

Zoom by touch panel

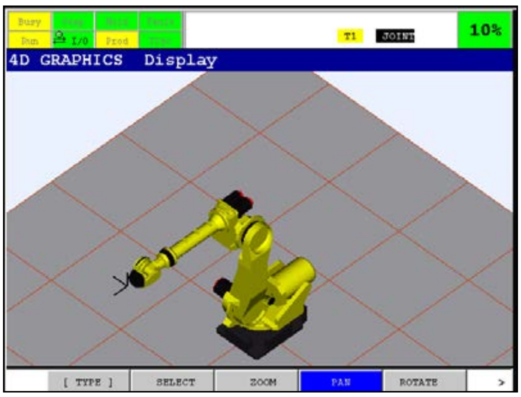
Increase the magnification

- Touch and release near the top of the screen.
- Touch the screen and drag your finger up or right,

Decrease the magnification

- Touch and release near the bottom of the screen.
- Touch the screen and drag your finger down or left,

PAN



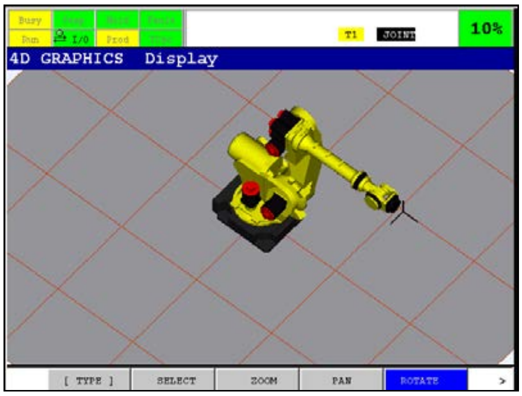
PAN consists of moving the view up, down, left and right. Press F4, PAN to set the system to PAN mode. Then, the label F4, PAN turns blue in color.

There are two ways for PAN operation. One is the *i*Pendant key input and the other is touch panel operation.

PAN by *i*Pendant key input
- Press the arrow key on iPendant up, down, left and right.

PAN by touch panel
- Touch the screen and drag your finger up, down, left and right.
- Click the screen so the point you click become center on the screen.

ROTATE



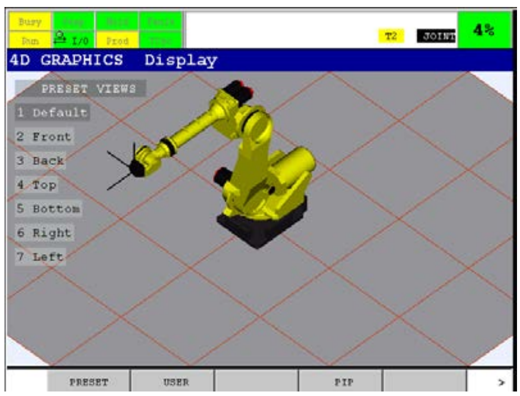
ROTATE consists of rotating the view up, down, left and right. Press F5, ROTATE to set the system to ROTATE mode. Then, the label F5, ROTATE turns blue in color.

There are two ways for ROTATE operation. One is the *i*Pendant key input and the other is touch panel operation.

ROTATE by iPendant key input
- Press the arrow key on *i*Pendant up, down, left and right.
- Press the SHIFT + arrow key on *i*Pendant up, down, left and right. (High magnification)

ROTATE by touch panel
- Touch the screen and drag your finger up, down, left and right.

Preset Views



4D GRAPHICS screen provides seven preset views. The default view provides a view from 45 degree. This view is good starting point for setting the view. It also put information back on the screen in the case where it has inadvertently been lost. All preset views will center the floor in the middle of the view.
- Default Set the view to default
- Front Set the view in front of the robot
- BackSet the view behind the robot
- Top Set the view right above the robot

- Bottom Set the view right below the robot
- Right Set the view on the right side of the robot
- Left Set the view on the left side of the robot

To select Preset Views, press NEXT key and press F2, [VIEWS]. Select one of the preset views above.

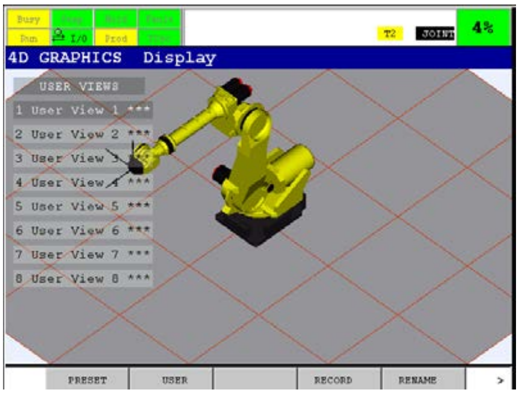
User Views

4D GRAPHICS screen provides eight user views. These views can be recorded and then retrieved by selecting them.

Procedure 7-7 Record User Views

Step

1 Move to the view you want to record.
2 Press the [NEXT] key and press F2, [VIEWS].
3 Press F2, [USER], so following screen will be displayed.



4 Select one of User Views you want to record and press F4, [RECORD]. When User View was recorded, "***" displayed the right of the User View was removed.

Once a User View is recorded, it is available to be retrieved at any time.
Press F5, [RENAME] on one of User Views, then User View can be named.

SELECT and [VISIBLE]

F2, SELECT and F3, [VISIBLE] after pressing the [NEXT] key are optional functions.

7.7.2 Current Position

The current position of the robot shows the location and the orientation of the robot in the work space. The current position can be represented in the Cartesian frame and the joint frame.

Joint coordinates

Joint coordinates represent the current position by the angular displacement from the base side of each axis.

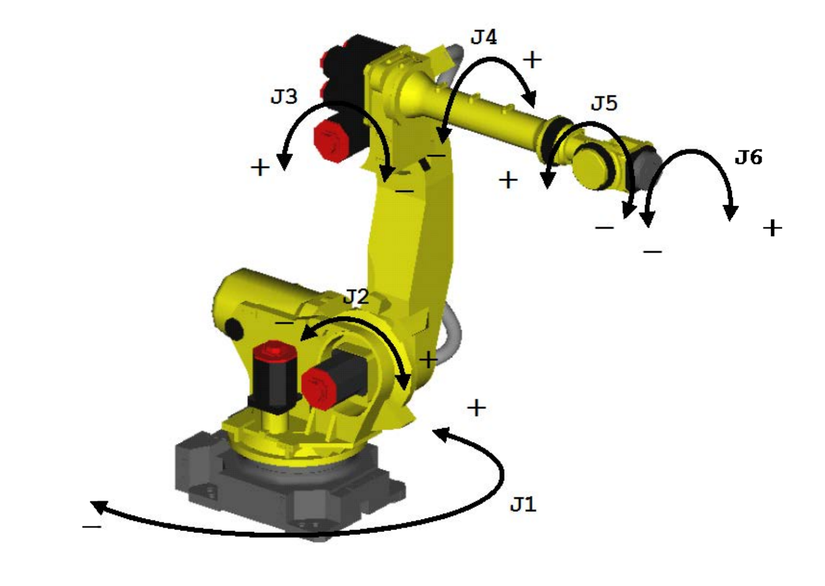


Fig. 7.7.2 (a) Joint coordinate system

Displaying joint coordinates

```
POSITION
     Joint                            Tool: 1

     J1:    90.045 J2:    42.195 J3:     29.336
     J4:    40.000 J5:    10.000 J6:     20.000
     E1:    11.942 E2:     ***** E3:      *****
     J2/J3 Interaction:      71.531

     (J7) G1 / J7                   11.942
     (J8) G2 / J1                    3.780

   [ TYPE ]     JNT      USER     WORLD
```

> **NOTE**
> If the system has an additional axis, E1, E2 and E3 indicate the position data of the additional axis. If J7 key and J8 key are set, (J7) and (J8) display the position data of the axis which is assigned to each key. Refer to “Setting of J7 key and J8 key” in the Subsection 5.2.3 Moving the Robot by Jog Feed about the setting for J7 and J8 key.

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.

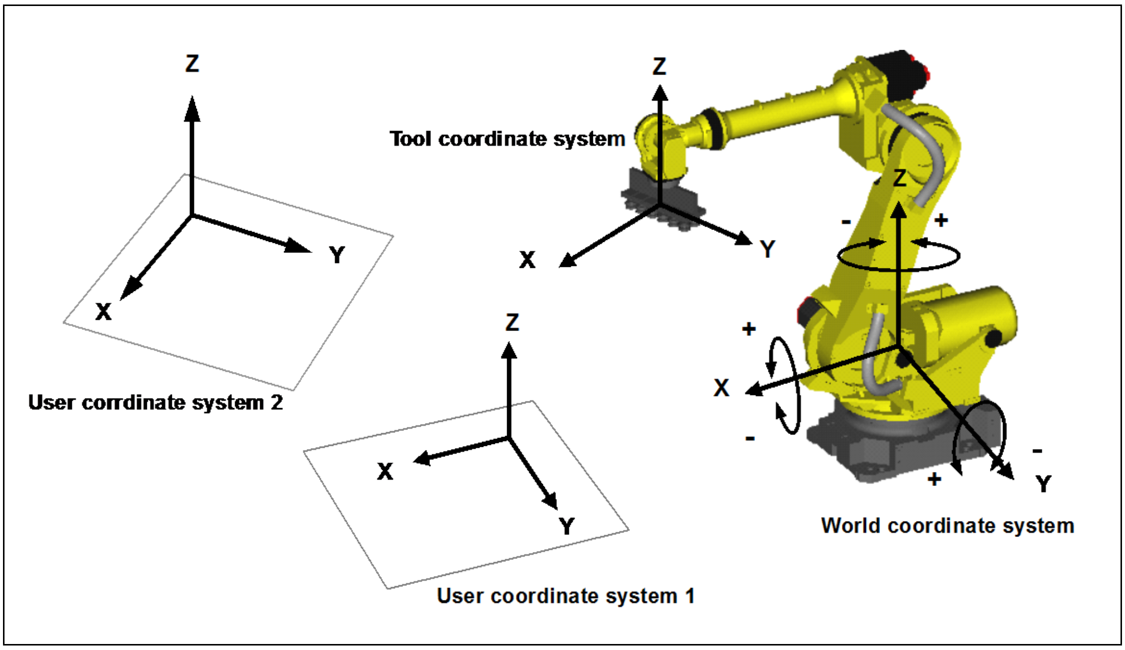


Fig. 7.7.2 (b) Cartesian coordinate system

Displaying world coordinate system

```
POSITION
      World                        Tool: 1

     Configuration: N U T, 0, 0, 0
      X:  1380.000  y:  -380.992  z:  956.895
      W:    40.000  P:   -12.676  r:   20.000
     E1:    11.942 E2:     ***** E3:    *****


    (J7) G1 / J7                 11.942
    (J8) G2 / J1                  3.780

   [TYPE]   JNT     USER    WORLD
```

Displaying user coordinate system

```
POSITION
      User             Frame: 0  Tool: 1

     Configuration: N U T, 0, 0, 0
      X:  1500.374  y:  -342.992  z:  956.895
      W:    40.000  P:    10.000  r:   20.000
     E1:    11.942 E2:     ***** E3:    *****


    (J7) G1 / J7                 11.942
    (J8) G2 / J1                  3.780

   [TYPE]   JNT     USER    WORLD
```

Procedure 7-8 Displaying current position screen

Step

1. Press [MENU] key to display the screen menu.
2. Select NEXT, then select POSITION from the next menu.
3. The current position screen can be also displayed by pressing the POSN key.
   - To display joint coordinates, press F2, JNT.
   - To display user coordinates, press F3, USER.
   - To display world coordinates, press F4, WORLD.

7.8 SYSTEM VARIABLES

All the system variables can be seen with the system variable screen. Settings of the system is stored in the system variables.

> ⚠ **WARNING**
> The operation of the robot and controller is controlled with system variables. Only a person who knows details of the influence of changes in system variables should set system variables. If a person without detailed knowledge attempts to set the system variables, the robot and controller would malfunction.

Procedure 7-9 Displaying system variable screen

Step

1. Press [MENU] key. The screen select menu will be displayed.
2. Select NEXT, then select SYSTEM.
3. Press F1, [TYPE].
4. Select Variables. The system variable screen will be displayed.

```
SYSTEM Variables
                                       1/665
     1 $AAVM_GRP          AAVM_GRP_T
     2 $ABSPOS_GRP        ABSPOS_GRP_T
     3 $ACC_MAXLMT        150
     4 $ACC_MINLMT        0
     5 $ACC_PRE_EXE       0
     6 $ACC_UPDATE        *uninit*
     7 $ALM_IF            ALM_IF_T
     8 $ANGTOL            [9] of REAL
     9 $APPLICATION       [9] of STRING[21]
    10 $AP_ACTIVE         6
    11 $AP_AUTOMODE       FALSE

   [ TYPE ]  DETAIL
```

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 keys.
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 will be displayed.

> ⚠ **WARNING**
> Cycling power is required to make a new setting valid. Otherwise, injury or property damage would occur.

```
SYSTEM Variables
                                        350/665
    345 $OVRD_RATE        5
    346 $OVRD_SETUP       OVRD_SETUP_T
    347 $PADJ_SCHNUM      10
    348 $PALCFG           PALCFG_T
    349 $PARAM2_GRP       MRR2_GRP_T
    350 $PARAM_GROUP      MRR_GRP_T
    351 $PARAM_MENU       [21] of STRING[21]
    352 $PASSNAME         [10] of PASSNAME_T
    353 $PASSSUPER        PASSNAME_T
    354 $PASSWORD         PASSWORD_T
    355 $PAUSE_PROG       *uninit*

   [ TYPE ]  DETAIL
```

```
SYSTEM Variables
    $PARAM_GROUP[1]                       1/236
      1 $BELT_ENABLE      FALSE
      2 $CART_ACCEL1      800
      3 $CART_ACCEL2      400
      4 $CIRC_RATE        1
      5 $CONTAXISNUM      0
      6 $EXP_ENBL         FALSE
      7 $JOINT_RATE       1
      8 $LINEAR_RATE      1
      9 $PATH_ACCEL1      800
     10 $PATH_ACCEL2      400
     11 $PROCESS_SPD      2000.0

   [ TYPE ]                    TRUE     FALSE
```

7 To return to the upstairs layer, press the [PREV] key.

7.9 PROGRAM TIMER

A program timer is a timer for measuring the execution time from one line to another in a program. Ten program timers can be used as standard.
A program timer can be started and stopped by using a timer instruction (see Subsection 4.15.3). It also stops at forced termination and upon a halt.

There are two modes program timer, local timer and global timer.

- Local Timer
  Local timer is stopped at forced termination and upon a halt. Timer count is restarted when program is resumed after a halt.
- Global Timer
  Global timer does not depend on state of program. Timer is not stopped at forced termination and upon a halt. Global timer can be measured time including stop time and execution time of some programs.
  Timer mode is set in the program timer detail screen.

Timer flag output signal by timer value.
  Timer count value < 0 : OFF
  Timer count value >= 0 : ON
Output signal port is set in the program timer detail screen. DO, RO and F can be used.
Example:
  When DO[1] is set as timer flag, if TIMER[1]<0, DO[1] is OFF, if TIMER[1]>=0, DO[1] is ON.

The program timer detail screen displays the following information:

- Setting of Program timer mode (Local or Global)
- Program name and line number for which a timer was started most recently
- Program name and line number for which a timer was stopped most recently
- Setting of Timer Flag

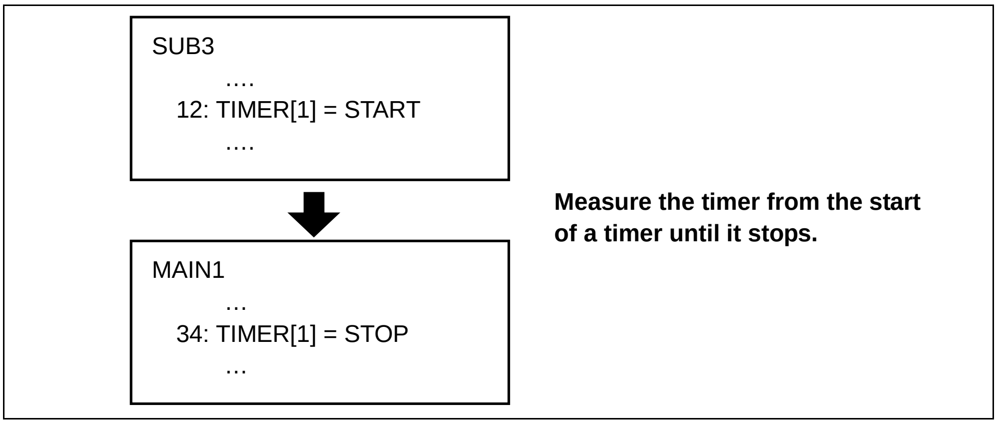


Fig. 7.9 Program timer measurement

Program timers are indicated by using [4 STATUS Prg Timer] on the program timer screen.

Procedure 7-10 Displaying program timers

Step

1. Press [MENU] 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 will be displayed.

```
PRG TIMER LISTING
                                           1/20
    TIMER   count          comment
      1    123.45    (s)[TIMER TEST     ]
      2 G    0.00    (s)[               ]
      3      0.00    (s)[               ]
      4      0.00    (s)[               ]
      5      0.00    (s)[               ]
      6      0.00    (s)[               ]
      7      0.00    (s)[               ]
      8      0.00    (s)[               ]
      9      0.00    (s)[               ]
     10      0.00    (s)[               ]

   [ TYPE ]  DETAIL
```

5. If timer mode is set global timer, ‘G’ is displayed beside Timer number.
6. To display detail information, press F2, DETAIL. Then, the program timer detail screen will be displayed.

```
PRG TIMER DETAIL
                                        1/3
    Timer[ 1]
     Comment       :[TIMER TEST    ]
     Count         :       123.45(sec)
     Local/Global  :LOCAL
     Start program :
      [TEST                              ]
            line   :         1
     Stop program :
      [TEST                              ]
            line   :         3
     Timer flag    :DO  [    0]

   [ TYPE ]  LISTING
```

7 To enter a comment, position the cursor to the comment field, and press the [ENTER] key. Select the input method, and enter characters using function keys.
8 As the start program, a program for which the timer was started most recently is indicated.
As the stop program, a program for which the timer was stopped most recently is indicated.
9 To set program timer mode, position the cursor to the Local/Global field, and press F4, Global or F5, Local. If timer mode is changed while running, timer is stopped.
10 To set output port of timer flag, position the cursor to the timer field. To select kind of output signal, press F3, [CHOICE] and select from DO, RO or F. To input port index, position the cursor to the index number and input number of index. If port index of timer flag is 0, timer flag is disabled.

NOTE
Timer flag setting is enabled after next timer start. If timer flag setting is changed while timer is running, timer flag setting is disabled until timer is stopped.

7.10 SYSTEM TIMER

A system timer is a timer for indicating the system operation time. The times for five items are indicated. Five types of timers are provided for each operation group.

Table 7.10 System timer display

Item	Description
On power time	Time during which the power to the controller is on.
Servo on time	Time during which the system is ready for operation (servo on) after the release of an alarm.
Running time	Program execution time. The halt period is not included.
Waiting time	Time required to execute a WAIT instruction.
Mech Unt time	Time during which servo brake is released.

To display the system timers, use [4, STATUS Sys Timer] on the system timer screen.

Procedure 7-11 Displaying system timer screen

Step

1 Press [MENU] 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 will be displayed.

```
SYS TIMER
                                              1/4

        GROUP : 1
          Timer type  Total(h)  Lap(m)
        On Power time:    12.3     0.0[OFF]
        Servo on time:     4.5     0.0[OFF]
        Running  time:     2.3     0.0[OFF]
        Waiting  time:     1.2     0.0[OFF]
        Mech Unt time:    0.8      0.0[OFF]

   [ TYPE ]  GROUP#   ON/OFF   RESET
```

5 To switch between operation groups, press F2, GROUP#, and enter a group number.
6 To enable or disable lap time measurement, position the cursor to a desired item, and press F3, ON/OFF to switch the setting.
7 To reset the lap time, position the cursor to a desired item, and press F4, RESET.

7.11 EXECUTION HISTORY

The function of the program execution history records the execution history of the program which has been executed or which is being executed at the end, and enables you to see the execution history after the program is finished or paused.
For example, this function enables you to recognize the execution status of the program at power failure after the cold start is done in case that power supply is turned off for any causes while the program is executed.

> **NOTE**
> You can not see the execution history of the program which is been executed.

The following information can be referred with the execution history screen.

- Executed program name and line number (The status of the latest executed program is displayed at the first line.)
- Direction of execution
  - FWD : The line was executed by the forward execution.
  - BWD : The line was executed by the backward execution.
- Status of execution
  - Not exec : The line was read but the line has not been executed.
  - Paused : (The program was paused while executing the line.)
  - Done : The execution of the line has been completed.
  - Aborted : The program has finished to be executed.

The maximum number of the execution history which can be recorded is 200. The number of record lines can be changed using the maximum number setting screen, selectable from the controlled 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 7-12 Displaying program execution history

Step

1 Press [MENU] key. The screen select menu will be displayed.
2 Select STATUS from the next page.
3 Press F1, [TYPE].
4 Select Exec-hist. The execution history screen will be displayed.

```
Execution history
                                              1/5
        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




    Press NEXT to display other task

   [ TYPE ]   NEXT             ALL_CLR   CLEAR
```

> **NOTE**
> If a single program has been executed, F2, NEXT and F4, ALL_CLR are not displayed on the execution history screen.

5 Only when the displayed status of a program is ”Aborted”, the execution history can be cleared by pressing SHIFT + F5, CLEAR.
6 When multitasking is used, pressing F2, NEXT displays the history of another task.
7 When multitasking is used, the execution history of all tasks can be cleared by SHIFT + F4, ALL_CLR, only if the displayed status of all tasks are “Abort”.

7.12 MEMORY USE STATUS DISPLAY

This screen displays the use status and hardware configuration of the controller memory. The display includes the following information:

Table 7.12 (a) Memory use status display (pools)

Items	Descriptions
TPP	Displays the use of area to hold programs.
PERM	Displays the use of area to hold system variables and registers.
TEMP	Displays the use status of work area used by system software.

Table 7.12 (b) Memory use status display (hardware)

Items	Descriptions
F-ROM	Storage capacity of the F-ROM module used in controller
D-RAM	Storage capacity of the D-RAM (RAM) module used in controller
C-MOS	Storage capacity of the C-MOS (RAM) module used in controller

When the [STATUS Memory] screen is selected, the following screen will be displayed 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
                   Total        Available
       Pools      ------------------------
         TPP       1000.0KB        886.5KB
         PERM      3002.0KB       1508.3KB
         TEMP     22164.4KB      10561.9KB
         FR       31857.0KB      17502.0KB

        Description:
        TPP:   Used by .MN, .MR, .JB, .PR
        PERM:  Used by .VR, SV, 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 detail screen

```
STATUS Memory
              Total        Free    Lrgst Free
    Pools    ---------------------------------
     TPP      1000.0KB    986.5KB     985.8KB
     PERM     3002.0KB   1508.3KB    1505.3KB
     SYSTEM   6058.0KB      2.6KB       2.6KB
     SHADOW   6430.4KB   6229.9KB      32.0KB
     TEMP    22164.4KB  10562.0KB    5928.7KB
     FR      31857.0KB  17502.0KB
    Hardware --------------------------------
     FROM      32.0 MB (T) DRAM        32.0 MB
     SRAM       3.0 MB (C-MOS)

   [ 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 the [PREV] key.

> **NOTE**
> This function indicates the use status of the memory. It does not change the use status.

7.13 STOP SIGNAL

The stop signal screen indicates the state of safety related signals.
To be specific, the screen indicates whether each stop signal is currently on. It is impossible to change the state of any stop signal on this screen.

Table 7.13 Stop signals

Stop Signal	Description
SOP E-Stop	This item indicates the state of the emergency stop button on the operator's panel. If the EMERGENCY STOP button is pressed, the state is indicated as "TRUE". The state of the EMERGENCY STOP button on the operator's panel cannot be determined while the EMERGENCY STOP button on the teach pendant is pressed. As a result, the state is indicated as "FALSE".
TP E-Stop	This item indicates the state of the emergency stop button on the teach pendant. If the EMERGENCY STOP button is pressed, the state is indicated as "TRUE".
Ext E-Stop	This item indicates the state of the external emergency stop signal. If the EMERGENCY STOP signal is asserted, the state is indicated as "TRUE".
Fence Open	This item indicates the state of the safety fence. If the safety fence is open, the state is indicated as "TRUE".
TP Deadman	This item indicates whether the DEADMAN switch on the teach pendant is grasped. If the teach pendant is operable, and the DEADMAN switch is grasped correctly, the state is indicated as "FALSE". If the DEADMAN switch is released or grasped tightly when the teach pendant is operable, the state is indicated as "TRUE". In this case, an alarm occurs, causing the servo power to be switched off.
TP Enable	This item indicates whether the teach pendant is operable. If the teach pendant is operable, the state is indicated as "TRUE".
Hand Broken	This item indicates the state of the hand safety joint. If the hand interface with a workpiece or anything like this, and the safety joint is opened, the state is indicated as "TRUE". In this case, an alarm occurs, causing the servo power to be switched off.
Overtravel	This item indicates whether the current position of the robot is out of the operation range. If any robot articulation goes out of the operation range beyond the overtravel switch, the state is indicated as "TRUE". In this case, an alarm occurs, causing the servo power to be switched off.
Low Air Alarm	This item indicates the state of the air pressure. The abnormal air pressure signal is connected to the air pressure sensor. If the air pressure is not higher than specified value, the state is indicated as "TRUE".
Belt Broken	This item indicates the state of the belt. If the belt broken robot digital input is asserted, the state is indicated as "TRUE".
SVOFF Input	This item indicates the state of the SVOFF (Servo Off Signal). If the SVOFF input signal connected to the panel board are open, the state is indicated as "TRUE".
Non Teacher Enb. Dev.	This item indicates the state of the NTED (Non Teacher Enabling Device) signal. If the NTED input signal connected to the panel board are open in the teach mode (T1, T2), the state is indicated as "TRUE".

NOTE
Refer to the "FANUC Robot series R-30*i*B/R-30*i*B Plus CONTROLLER MAINTENANCE MANUAL" (B-83195EN) for more information on each stop signal.

Procedure 7-13 Displaying stop signal screen

Step

1 Press [MENU] key. The screen select menu will be displayed.
2 Select "0 NEXT", then select "4 STATUS".
3 Press F1, [TYPE] to display the screen switching menu.
4 Select Stop Signal. Then, the following screen will be displayed.

```
STATUS Stop Signal
         SIGNAL NAME          STATUS    1/12

      1 SOP E-Stop:             TRUE
      2 TP E-Stop:              TRUE
      3 Ext E-Stop:             TRUE
      4 Fence Open:             TRUE
      5 TP Deadman:             TRUE
      6 TP Enable:              TRUE
      7 Hand Broken:            TRUE
      8 Overtravel:             TRUE
      9 Low Air Alarm:          TRUE
     10 Belt Broken:            TRUE
     11 SVOFF Input:            TRUE
     12 Non Teacher Enb. Dev.: TRUE

  [ TYPE ]
```

7.14 PROGRAM STATUS

The program status screen display information about the running teach pendant or KAAREL program, and the running routine if a KAREL program is running. Table 7.14 lists and describes each Program status item.

Use Procedure 7-14 to display the program status screens.

Procedure 7-14 Displaying program status screen

Step

1 Press [MENU] key.
2 Select STATUS.
3 Press F1, [TYPE].
4 Select Program. The program status screen will be displayed. The following screen will be displayed.

```
STATUS Program
                                          1/4
      1 Task number:  7
      2 Task name:    RSR0001
      3 Program:      MAIN
      4 Routine:      MAIN
        Line number:  6
        Status:       RUNNING

  [ TYPE ]   PREV    NEXT
```

If you press F2, PREV, previous task number status program screen will be displayed.
If you press F3, NEXT, next task number status program screen will be displayed.

Table 7.14 Program status items

Item	Description
Task number	This item displays number of task. If you change this item, overall screen is changed with this item.
Task name	This item displays name of main program that started execution.
Program	This item displays the name of the teach pendant or KAREL program that is currently being executed.

Item	Description
Routine	This item displays the name of the KAREL routine that is being executed, if there is a KAREL program currently being executed.
Line number	This item indicates the line of the teach pendant or KAREL program that is currently being executed.
Status	This item displays the status of the teach pendant or KAREL program that is currently being executed. The status of a running program can be ● RUNNING ● PAUSED ● ABORTED ● MOVING ● WAITING ● WAITING FOR DI[n] ● WAITING FOR RI[n]

7.15 POWER CONSUMPTION MONITOR

This screen displays the supplying power and the power that is regenerated from regenerative unit (The regenerative unit is option hardware).

Table 7.15 Items for power consumption monitor

Items	Descriptions
Supplying Power	Displays power consumption without the regenerative unit.
Regenerative Power	Displays power that is regenerated from the regenerative unit.
Current Power	Displays the value obtained by subtracting Regenerative Power from Supplying Power.

NOTE
The powers displayed in this screen are the results calculated by simulation and there are errors by comparison with actual power consumption.

Procedure 7-15 Displaying power consumption monitor screen

Step

1. Press [MENU] key. The screen select menu will be displayed.
2. Select STATUS from the next page.
3. Press F1, [TYPE].
4. Select Power Consumption. The Power Consumption Monitor screen will be displayed.

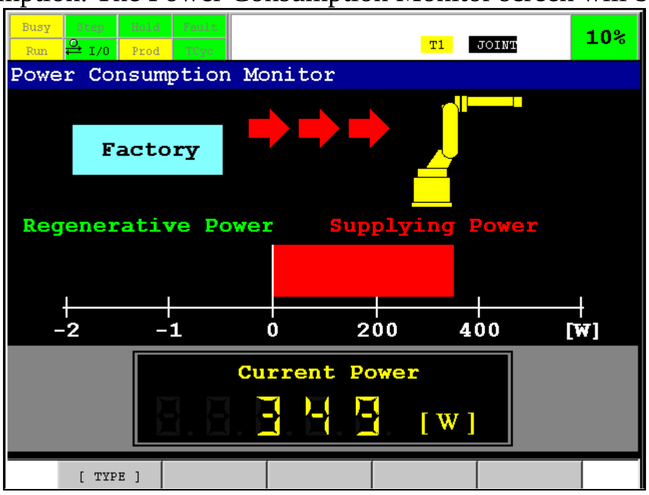


Fig. 7.15 Power consumption monitor screen

NOTE

If there is at least one robot which is not supported by this function, the following message is displayed.

"PCM does not support this robot model."

If the robot is supported by this function, but the extended axis or the positioner is not supported, only the power of the robot is displayed.

7.16 NOTIFICATIONS

The Notifications screen displays the current list of controller issues to be addressed. You may have no issues or you may have different issues based on your system.

If you see a blinking icon on the Status Bar, hereafter called Notification Icon, please confirm the notification on the notification screen. If you can respond to the issue, select the description for help. Once the issue has been resolved, the notification will be cleared automatically.

To clear Notification Icon, please select the clear button, select a term you would like to clear the notification, and select OK. Then, the notification will be cleared temporarily.

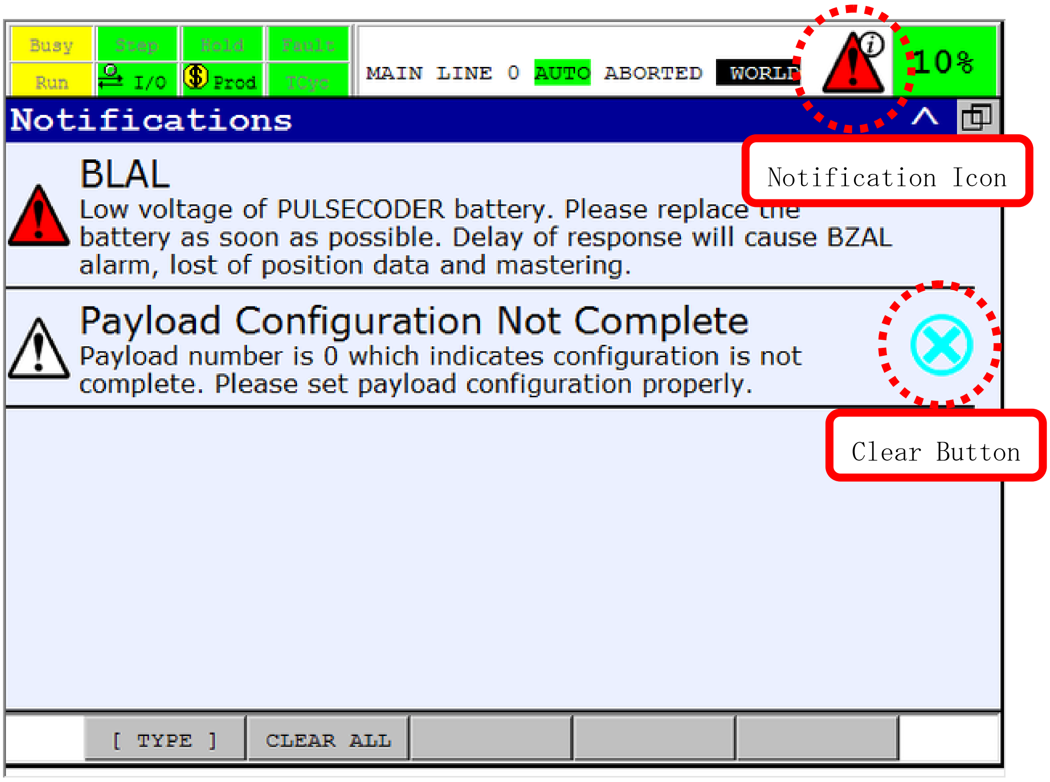


Fig. 7.16 (a) Notifications screen

Procedure 7-16 Displaying Notifications screen

There are 3 ways to display the Notifications screen.

- If Notification Icon is displayed on the Status Bar, you press the icon.
- If Notification Icon is displayed on the Status Bar, you press and release ***i*** key.
- If no Notification Icon is displayed, you can press MENU, Select STATUS, Press F1 [TYPE], Select Notifications.

NOTE

Without touch panel option, Notifications screen cannot be displayed by pressing Notification Icon.
With touch panel option, pressing button on Notifications screen can be replaced by tapping button on touch panel.

Table 7.16, “Notifications Screen Items” lists and describes each item on the Notifications screen.

Table 7.16 Notifications Screen Items

Items	Descriptions
Warning Icon	The color of the warning icon indicates its severity. • Red: Alert level notification • Yellow: Caution level notification • White: Low level notification • Gray: Cleared notification The notifications are displayed in order with Red at the top and Gray at the bottom.
Title and Description	Typically the title and description is shown on a button which when pressed, either displays a message or switches to the menu which can be used to fix the issue.
Clear Button	The clear button displays a dialog to clear the notification for a day or for a week. When cleared, the notification changes to a gray color. It remains cleared until either the issue is fixed, the time elapses, the notification is restored, or the notification level changes to a higher level. Some notifications are not allowed to be cleared, such as BLAL.
Restore Button	The restore button displays a dialog to restore a cleared notification.
CLEAR ALL	This function key displays a dialog to clear all the notifications that have a clear icon. It also resets the time for any previously cleared items.
Notification Icon on Status Bar	The color of Notification Icon indicates the highest severity of all notifications, either red, yellow, or white. Cleared notifications will not cause Notification Icon to display. The blinking can be controlled by the $UI_CONFIG.$BLINK_ICON system variable. This is a timer in msec in which the icon will blink. To prevent the icon from blinking, you may set it to 0.

Figure 7.16(b), “Notifications Screen with Cleared Notification” shows the Notification screen after clearing the payload configuration issue temporarily.

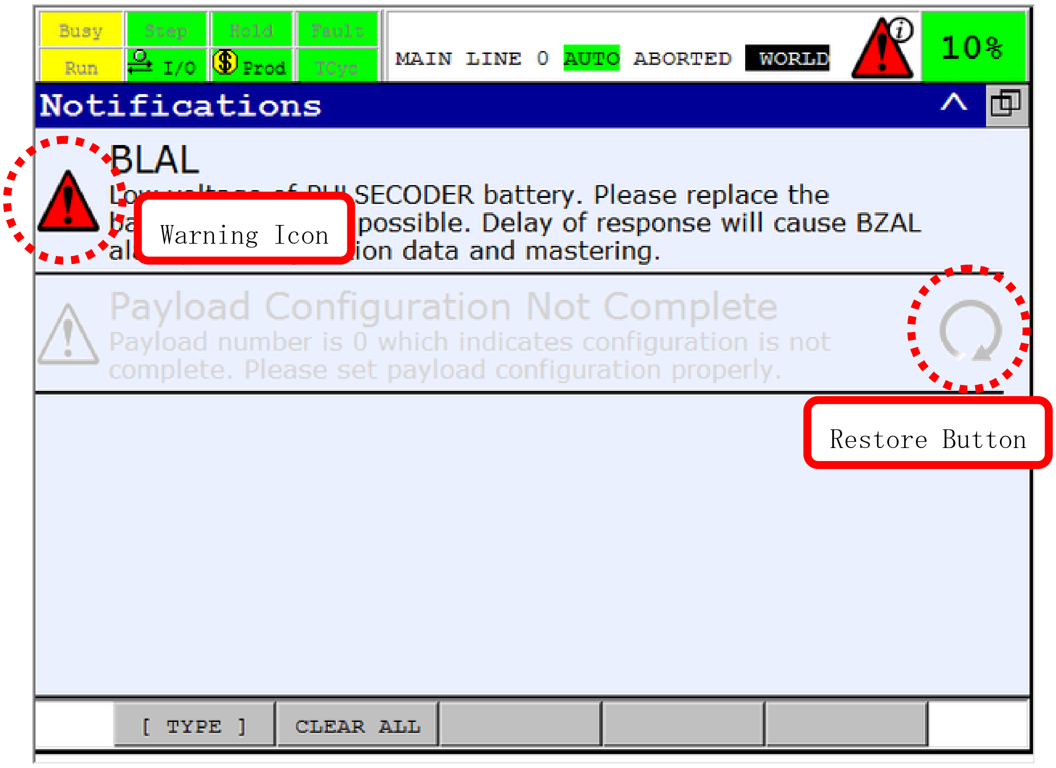


Fig. 7.16 (b) Notifications Screen with Cleared Notification

Notifications can also be viewed from the robot HOME page, using Error/Diagnostic files available on MD: and selecting NOTIFY.DG

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 controller, the various file I/O devices can be used.
The standard setting specifies the use of memory cards on R-30*i*B or R-30*i*B Plus controllers, and specifies the use of USB memory on R-30*i*B Mate, R-30*i*B Mate Plus, R-30*i*B Compact Plus or R-30*i*B Mini Plus controllers. If using another file I/O device, perform the operation below to change the file I/O device setting.

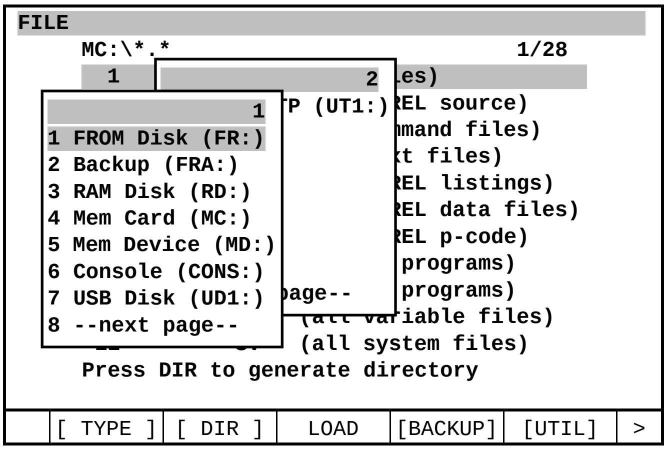
Procedure 8-1 Changing file I/O devices

Step

1 Press [MENU] key to display the screen menu.
2 Select 7 FILE. The file screen will be displayed.

```
FILE
     MC:\*.*                              1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Press DIR to generate directory

   [ TYPE ] [ DIR ]   LOAD   [BACKUP] [UTIL]  >
```

3 Press F5, [UTIL], and select Set Device. Then, the following screen will be displayed :



4 Select a file I/O device to be used. An abbreviation for the currently selected file I/O device will be displayed in the upper left part of the screen.

File input/output units

The Robot controller allows the use of the following types of storage units to save programs and files.

- Memory card (MC:)
  Flash ATA memory card. It is possible to use a Compact Flash card by attaching a PCMCIA adapter to it. The memory card slot is on the main board of R-30*i*B or R-30*i*B Plus controllers. On R-30*i*B Mate, R-30*i*B Mate Plus, R-30*i*B Compact Plus or R-30*i*B Mini Plus controllers, memory card interface is not available.
- Backup (FRA:)
  Area to which files are saved with auto backup. It can retain information when the power is interrupted, with no backup battery.
- FROM disk (FR:)
  Memory area which can retain information without backup battery. There are some important files for the system in the root directory of this storage device. It is possible to backup programs and save any file in this storage device. But, please do not save or delete files in the root directory. If you want to save file in this storage device, please create a subdirectory according to Procedure 8-20 and save the file in the created subdirectory.
- RAM disk (RD:)
  This is the storage device prepared for the special function. Please do not use this storage device.
- MF disk (MF:)
  This is the storage device prepared for the special function. Please do not use this storage device.
- FTP (C1: to C8:)
  Writes and reads files to and from a FTP server such as a PC connected via Ethernet. It is displayed only if FTP client settings have been made on the host communication screen.
- Memory device (MD:)
  The memory device is capable of handling data on the memory of a controller, such as robot programs and KAREL programs, as files.
- Console (CONS:)
  Device for maintenance only. It can reference the log file containing internal information.
- USB memory (UD1:)
  USB memory mounted to the USB port on the operator panel.
- USB memory (UT1:)
  USB memory mounted to the USB port on the teach pendant or Tablet TP.

The standard setting specifies the use of memory cards on R-30*i*B or R-30*i*B Plus controllers, and specifies the use of USB memory on R-30*i*B Mate, R-30*i*B Mate Plus, R-30*i*B Compact Plus or R-30*i*B Mini Plus controllers.

> ⚠ **CAUTION**
> The floppy disk (FLPY:) units below cannot be used.
> - Floppy cassette adapter (A16B-0150-B001)
> - Handy file (A16B-0159-B002)
>   (It cannot be used regardless of whether it is in FANUC or MS-DOS format.)
>
> The printer unit FANUC PRINTER (A86L-0001-0103) cannot be used. To print the content of a program, use the ASCII save function to output the content of the program in ASCII format to a file input/output unit, load it into a personal computer, etc., and print it.

> **NOTE**
> 1 When insert an USB memory to the tablet TP USB port, the directory “\Android\data\” is created automatically and some directories for android app are created automatically under “data” directory. This is the specification of Android OS.
> 2 Tablet TP app creates “\jp.co.fanuc.robot.tablettp\files\” directory under the “data” directory. When select UT1: on file screen of tablet TP, files and directories under the “files” directory are displayed.
> 3 When backup files to the root directory of UT1: on file screen of the tablet TP, files are saved under the “\Android\data\jp.co.fanuc.robot.tablettp\files” in the USB memory.
> 4 You cannot access the directories upper the “files” directory from file screen on the tablet TP.

Setting up a communication port

The Robot controller provides the communication ports below.

R-30*i*B and R-30*i*B Plus controllers
- Port 1 RS232C Operator panel
- Port 2 RS232C JD17 connector on the main CPU printed circuit board

R-30*i*B Mate, R-30*i*B Mate Plus, R-30*i*B Compact Plus and R-30*i*B Mini Plus controllers
- Port 1 RS232C JRS27 connector on the main CPU printed circuit board

The "Handy File", "FANUC Floppy", "Handy F MS-DOS", "Printer" items are not displayed as communication unit settings on the port setup screen.

Format

Some file I/O device can be formatted. Following is the procedure to format file I/O device.

Procedure 8-2 Formatting file I/O device

Step

1. Perform “Procedure 8-1 Changing file I/O devices” to select the desired file I/O device.
2. Press F5, [UTIL] and select “Format” or “Format FAT32”. In case that “Format” is selected, the file I/O device is formatted by FAT16 format. In case that “Format FAT32” is selected, the file I/O device is formatted by FAT32 format. The following screen is displayed.

```
FILE Format
     MC:\*.*                                1/28

              Formatting MC:

     ************   WARNING   ******************
             ANY DATA ON THE DISK WILL BE LOST!


      Format disk?

                              YES      NO
```

3 The confirmation message is displayed. Confirm whether the selected file I/O device is correct, and press F4, YES if file I/O device is correct.

4 The following screen is displayed.

```
FILE Format
     MC:\*.*                            1/28

                                   Alpha input 1
                                     Words
                                     Upper Case
     Enter volume label:             Lower Case
                                     Options
      PRG     MAIN     SUB     TEST
```

> ⚠ **CAUTION**
> All data in the file I/O device are lost by formatting. Please check the device carefully before formatting.

5 Enter the volume label using the function keys and the cursor keys. After the volume label is entered, the following screen is displayed, and the formatting is started.

```
FILE Format
     MC:\*.*                            1/28


     Formatting disk: xxxx
      PRG     MAIN     SUB     TEST
```

8.1.1 Memory Card (on R-30*i*B or R-30*i*B Plus)

The following memory card can be used.

Type	Recommended product
Compact flash memory card	Compact Flash memory card available from FANUC. A02B-0281-K601, A02B-0213-K211, A02B-0213-K212, A02B-0213-K213, A02B-0213-K214, A02B-0213-K216, A02B-0213-K217
PC card adapter	PC card adapter available from FANUC. A02B-0236-K150 (PC card size) A02B-0303-K150 (Half size)

A02B-0236-K150 A02B-0303-K150

Fig. 8.1.1 (a) PC card adapter

> ⚠ **CAUTION**
> 1 If a memory card other than those recommended is used, a normal operation is not guaranteed, and a bad influence may occur on the controller.
> 2 Please do not power off or remove memory card when writing or reading a file for not to break files in memory card.
> 3 Please re-format memory card when alarm "FILE-064 internal DOS system error:xx" is occurred because files in memory card may be broken.
> 4 A02B-0303-K150 is the compact PC card adapter which size is half of PC card, but note that this PC card adapter cannot be inserted into the card slot on the notebook PC.
> 5 On R-30*i*B Mate, R-30*i*B Mate Plus, R-30*i*B Compact Plus or R-30*i*B Mini Plus controllers, memory card interface is not available.

> ⚠ **CAUTION**
> It is recommended that files on a flash ATA memory card be backed up to storage device to protect the flash ATA memory card contents against accidental loss.

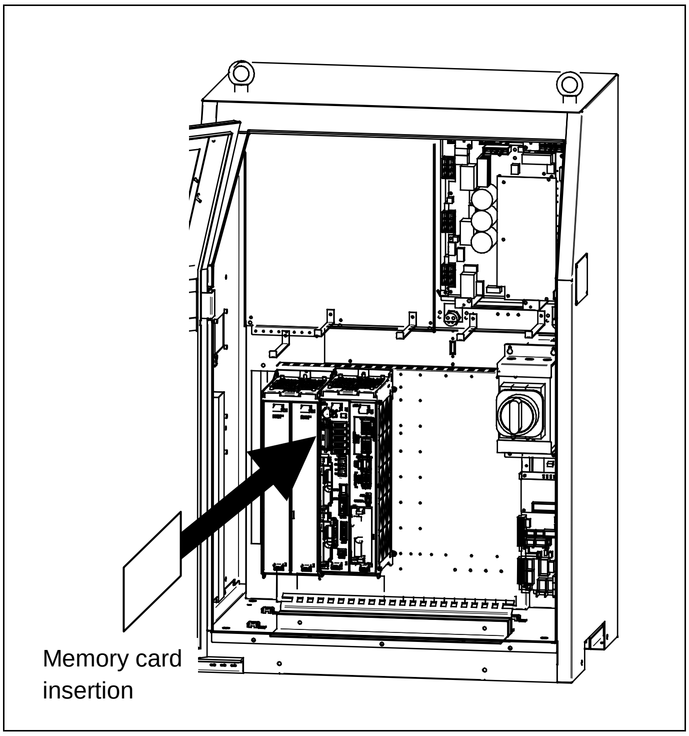


Fig. 8.1.1 (b) Memory card insertion

When a memory card is to be used, select the memory card according to the description of changing the file I/O devices (see Section 8.1).

8.1.2 USB Memory

Overview

The robot controller provides a USB port on its operation panel and the teach pendant, so that files can be loaded and saved using USB memory.

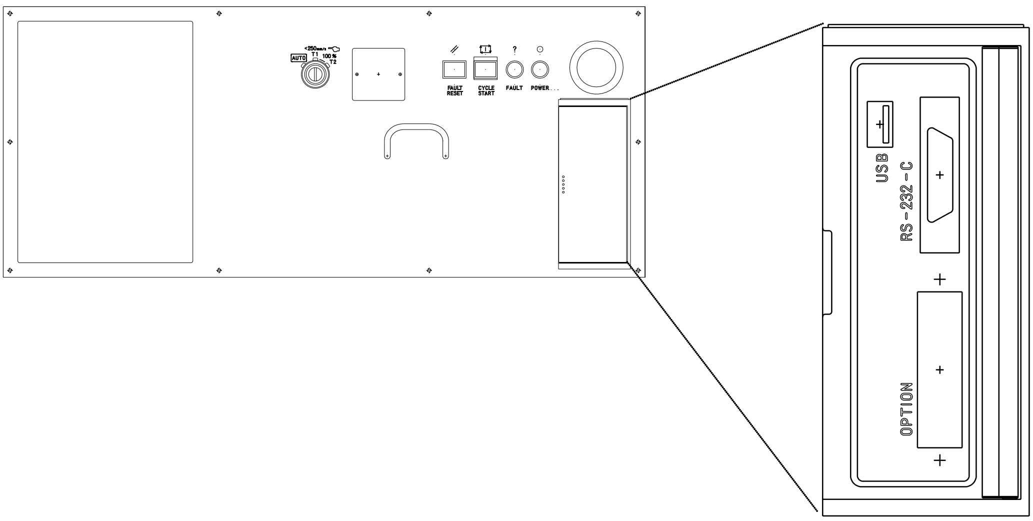


Fig. 8.1.2 (a) R-30*i*B USB port (Operation panel)

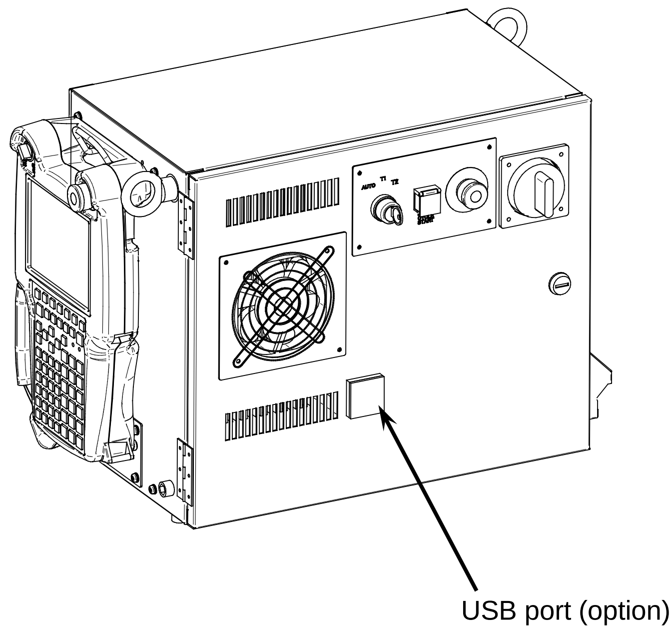


Fig 8.1.2 (b) R-30*i*B Mate USB port (Operation panel)

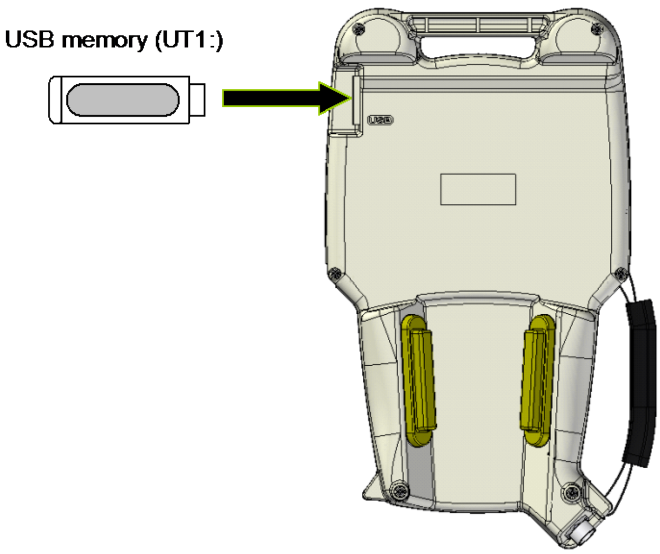


Fig. 8.1.2(c) USB port (Teach pendant)

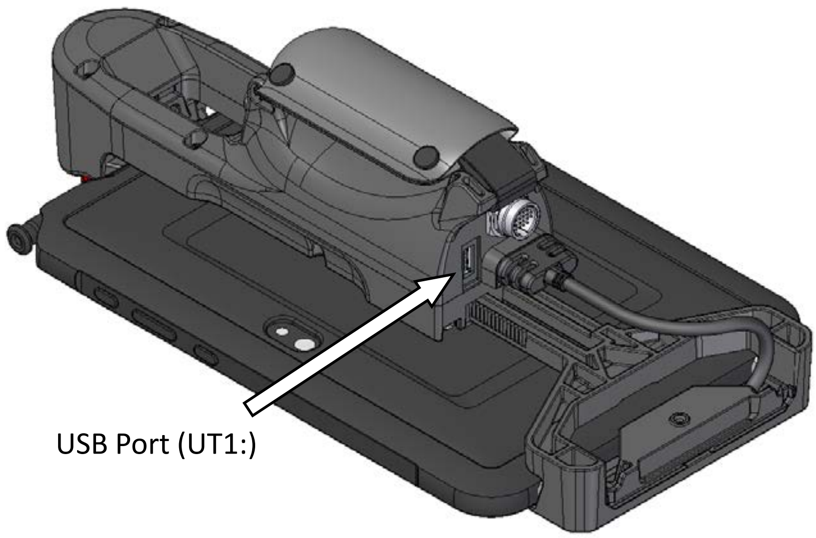


Fig. 8.1.2(d) USB port (Tablet TP)

> ⚠ **CAUTION**
> The USB memory units and the USB cameras which FANUC appointed are supported. Do not connect a USB unit other than them to the robot controller.

Procedure 8-3 Attaching a USB memory unit (Operation panel USB port)

Step

1. Insert a USB memory unit into the USB port on the operation panel.
2. If the USB memory unit is recognized correctly, the message below will be displayed on the alarm line on the teach pendant.

```
FILE-066  UD1 Ins BUFFALO ClipDrive
```

The alarm below is recorded in alarm history.

```
FILE-071  USB vend : a00e prod : 2868
```

"UD1 Ins" and subsequent text in the message above differ depending on the USB memory product. For example, if SanDisk Inc. Cruzer Micro is attached, the message below will be displayed.

```
FILE-066  UD1 Ins SanDisk Corporation Cruzer Micro
FILE-071  USB vend : 8107 prod : 5151
```

> ⚠ **CAUTION**
> 1 If the FILE-066 and FILE-071 message was not displayed even after a USB memory unit is attached, remove the memory unit and insert it again gently.
> 2 It is not possible to use two or more USB memory units at the same time.

3 Confirm the access lamp of USB memory is turned on and start to use it. If it is flashing, USB memory is not ready and wait to use until it is turned on.

> ⚠ **CAUTION**
> When a USB memory is used for the first time, format it on Robot controller.

> ⚠ **CAUTION**
> At every controller power up, the following alarms are recorded in alarm history whether USB memory is plugged in or not.
>
> **FILE-069 USB hub Ins On Time Informatik UHCI Virtual Root Hub**
> **FILE-071 USB vend:0000 prod:0000**
>
> When a USB memory is plugged in at controller power up time, FILE-066 and FILE-071alarms are recorded after above two alarms.

Procedure 8-4 Attaching a USB memory unit (Teach pendant USB port)

Step

1 Insert a USB memory unit into the USB port on the teach pendant.

> ⚠ **CAUTION**
> It is not possible to use two or more USB memory units at the same time.

2 When a USB memory unit is inserted into the USB port on the teach pendant, no message is displayed on the teach pendant. Confirm the access lamp of USB memory is turned on and start to use it. If it is flashing, USB memory is not ready and wait to use until it is turned on.

> ⚠ **CAUTION**
> When a USB memory is used for the first time, format it on Robot controller.

Procedure 8-5 Removing the USB memory unit (Operation panel USB port)

Step

1 Remove the USB memory unit from the USB port on the operation panel.
2 The message below will be displayed on the alarm line.

```
FILE-067  UD1 Removed
```

> ⚠ **CAUTION**
> While a file is being loaded or saved, do not remove the USB memory unit. Otherwise, the files in the USB memory unit may be damaged. If the USB memory unit is provided with an access lamp, make sure that the access lamp is not flashing before removing the unit.

Procedure 8-6 Removing the USB memory unit (Teach pendant USB port)

Step

1 Remove the USB memory unit from the USB port on the teach pendant.
2 When the USB memory unit is removed from the USB port on the teach pendant, no message is displayed on the teach pendant.

> ⚠ **CAUTION**
> While a file is being loaded or saved, do not remove the USB memory unit. Otherwise, the files in the USB memory unit may be damaged. If the USB memory unit is provided with an access lamp, make sure that the access lamp is not flashing before removing the unit.

Procedure 8-7 Switching to the USB memory unit (Common to operation panel USB port and teach pendant USB port)

The abbreviation of a USB memory unit inserted into the operation panel USB port as a file input/output unit is UD1:. And, the abbreviation of a USB memory unit inserted into the teach pendant or tablet TP USB port as a file input/output unit is UT1:.

Step

1 Press [MENU] key.
2 Select the FILE item to enter the file screen.
3 Press F5, [UTIL]. The following menu will be displayed.

```
      Function   1
1 Set Device
2 Format
3 Format FAT32
4 Make DIR
```

Select “Set Device”.

4 The following menu will be displayed.

```
                     1
1 FROM Disk (FR:)
2 Backup (FRA:)
3 RAM Disk (RD:)
4 Mem Card (MC:)
5 Mem Device (MD:)
6 Console (CONS:)
7 USB Disk (UD1:)
8 -- next page --
```

```
                     2
1 USB on TP (UT1:)
2
3
4
5
6
7
8 -- next page --
```

If you use USB memory unit inserted into the operation panel USB port, select “USB Disk (UD1:)”. If you use USB memory unit inserted into the teach pendant or tablet TP USB port, select “USB on TP (UT1:)”.

5 Check that the upper left file input/output unit currently selected is UD1: or UT1:.

```
FILE
     UD1:\*.*                          1/28
       1   *     *    (all files)
       2   *     KL   (all KAREL source)
       3   *     CF   (all command files)
       4   *     TX   (all text files)
       5   *     LS   (all KAREL listings)
       6   *     DT   (all KAREL data files)
       7   *     PC   (all KAREL p-code)
       8   *     TP   (all TP programs)
       9   *     MN   (all MN programs)
      10   *     VR   (all variable files)
      11   *     SV   (all system files)
     Press DIR to generate directory

   [ TYPE ] [ DIR ]   LOAD   [BACKUP]  [UTIL]  >
```

Procedure 8-8 Operating the USB memory unit (Common to operation panel USB port and teach pendant USB port)

The file operations below can be performed in the same way as those on the memory card (MC:).

- Load and save files on the program list screen
- Save files, display a file list, delete files, and load files on the file screen
- Format a USB memory unit on the file screen (Only the USB memory unit inserted into operation panel USB port)
- Save programs on the function menu

> ⚠ **CAUTION**
> 1 Please do not power off or remove USB memory when writing or reading a file for not to break files in USB memory.
> 2 Please re-format USB memory when alarm "FILE-064 internal DOS system error:xx" is occurred because files in USB memory may be broken.

> ⚠ **CAUTION**
> The USB memory units and the USB cameras which FANUC appointed are supported. Do not connect a USB unit other than them to the robot controller.

> **NOTE**
> It is not possible to format a USB memory unit inserted into the teach pendant USB port can not be form in the robot controller.

> **NOTE**
> The teach pendant does not have a real-time clock (RTC) circuit. For calendar function of the teach pendant, the robot controller sends the date and time to the teach pendant only at the time of power supply injection. Because of the software clock of teach pendant, the date and time of creation of file on UD1: is not correspond exactly to the date and time of the robot controller.

Available products

Some USB memory products cannot be recognized correctly by the robot controller or accept file operations correctly.

Those USB memory units that provide secure functions and require password authentication before access to the drive cannot be used.

Write protect notch of USB memory may not be functional.

Those USB memory units that have been confirmed for operation are as follows:

- RUF-C2GS/U2 of BUFFALO INC.
- TB-ST2/2G and TB-ST2/4G of I-O DATA DEVICE INC.
- TS4GJF350 of Transcend INC.

⚠ CAUTION

1. Those USB memory units above are confirmed for operation, but FANUC does not guarantee about USB memory on the market and does not accept responsibility to defective unit or malfunction by specification changes of device.
   Please confirm a USB memory unit at your site before you use it.
2. When a USB memory is used for the first time, format it on Robot controller.

8.2 SETTING A COMMUNICATION PORT

The controller performs data transfer to and from external devices through communication ports by performing serial communication via the RS232C interface.

The following communication ports are used. (Operator’s panel/box; see Subsection 2.3.2.)

R-30*i*B controller

- Port 1: RS232C On the operator’s panel
- Port 2: RS232C JD17 connector on the main CPU printed circuit board

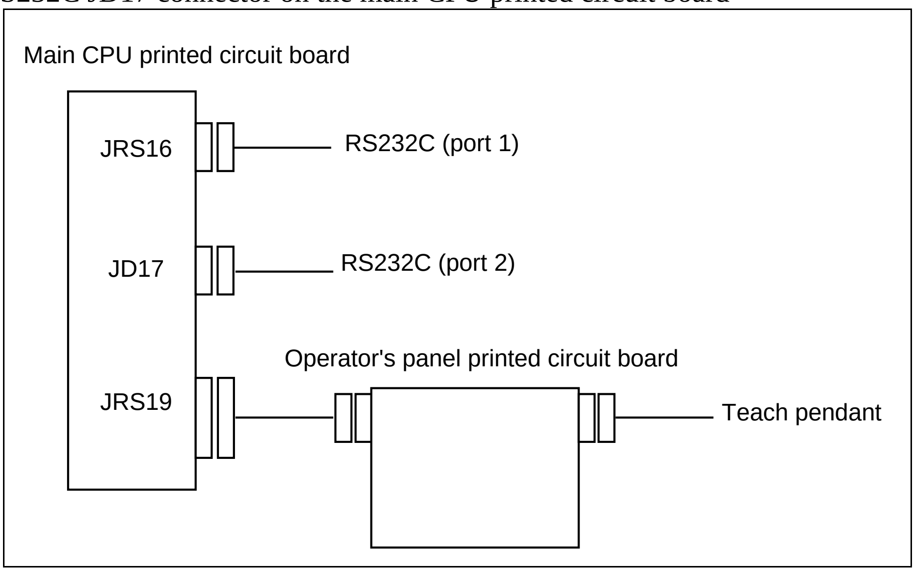


Fig. 8.2 (a) R-30*i*B Communication ports

R-30*i*B Mate controller

- Port 1: RS232C JRS27 connector on the main CPU printed circuit board

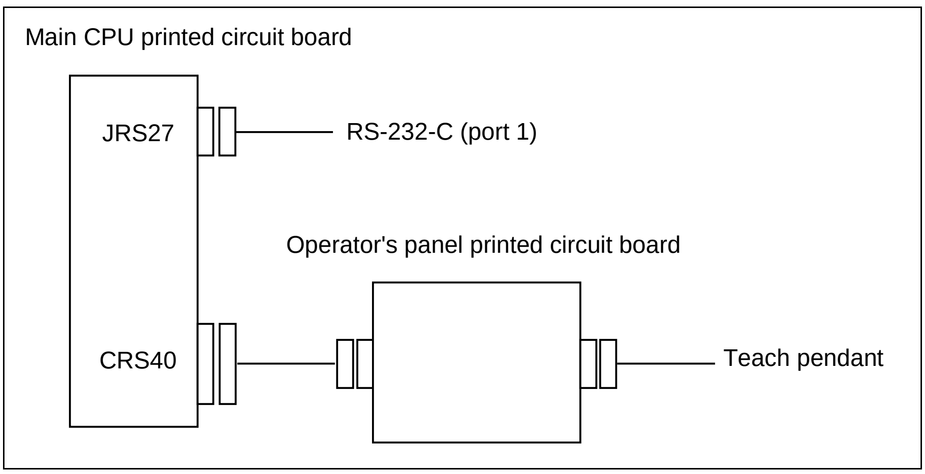


Fig. 8.2 (b) R-30*i*B Mate Communication ports

Communication ports are set by using [6 SETUP; Port Init] on the port setting screen.

Table 8.2 (a) Standard communication devices for communication ports

Communication port	Communication device
Port 1	Debug Console
Port 2	KCL/CRT

Table 8.2 (b) Setting a communication port

Items	Descriptions
Device	This item specifies a communication device to communicate with the robot controller. The standard communication devices that can communicate with the robot controller are listed below: • Host Comm Used when the R-30*i*B is connected to the host computer to use the data transfer function. • KCL/CRT • Factory Terminal • TP Demo Device **NOTE** When the communication device is changed, other settings such as a baud rate are changed to the corresponding standard values. Later on, the user can change each setting as desired.
Speed (Baud rate)	Baud rate is the transmission rate and it is the number of codes which can be transmitted per second. Enter the transmission rate specified for the peripheral unit being used.
Parity bit	To detect an error in data transfer, this item sets a mode of vertical parity check, which adds one extra bit to each transferred character. - Odd : The number of 1's in each transferred character must be an odd number. - Even : The number of 1's in each transferred character must be an even number. - None : No parity check is made. Enter the parity check mode specified for the peripheral unit being used.
Stop bit	This item specifies the number of stop bits to be added at the end of the transferred characters, for data transfer synchronization. - 1 bit : One stop bit is added. - 1.5 bits : One and a half stop bits are added. - 2 bits : Two stop bits are added. Enter the number of stop bits specified for the peripheral unit being used.
Time-out value (sec)	This item sets a maximum time during which control over transfer with a communication device must be exercised. If no data transfer occurs for a specified period of time, the communication line is disconnected.

Table 8.2 (c) Standard settings for communication devices

Device	Speed	Parity bit	Stop bit	Time-out value
Host Comm	9600	None	1 bit	None
Factory Terminal	9600	None	1 bit	None
KCL/CRT	9600	None	1 bit	None
TP Demo Device	9600	None	1 bit	None

Procedure 8-9 Setting a communication port

Step

1 Press [MENU] 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 will be displayed. (Only P1 port will be displayed on R-30*i*B Mate, R-30*i*B Mate Plus, R-30*i*B Compact Plus and R-30*i*B Mini Plus controllers.)

```
SETUP Port Init
                                        1/2
         Connector    Port     Comment
      1 JRS16 RS-232-C P2: [Maintenance Cons]
      2 JD17  RS-232-C P3: [KCL/CRT         ]

    [ TYPE ]          DETAIL
```

5 Move the cursor to a desired connecter port field, then press F3, DETAIL. The port setting screen will be displayed.

```
SETUP Port Init
     JRS16 RS-232-C P2                           1/5
       1 Device              [Maintenance Cons]
       2 Speed(Baud rate)                [9600 ]
       3 Parity bit                      [None ]
       4 Stop bit                      [1bit   ]
       5 Time out value(sec)              [   0]

    [ TYPE ]          LIST    [CHOICE]
```

6 To set a communication device, move the cursor to the “Device” field, then press F4, [CHOICE]. Select a desired communication device from the menu.

```
SETUP Port Init
     JRS16 RS-232-C P2                                1/5
       1 Devi                                     3   Cons]
                                             2  ,BP/Z)EQ1 600  ]
 1                      ent position ,BP/Z)EQ2 one  ]
 1 Sensor               programmer   ,BP/Z)EQ3 it   ]
 2 Host Comm                         ,BP/Z)EQ4 [   0]
 3 No Use               m/PPP        U
 4 KCL/CRT              device
 5 Maintenance Cons     Weld EQ1
 6 Factory Terminal     Weld EQ2      page --
 7 TP Demo Device       xt page --
 8 -- next page --

   [ TYPE ]              LIST    [CHOICE]
```

7 Select a communication device whose settings need to be changed. When the communication device is entered, the standard values are entered in the other setting fields.

```
SETUP Port Init
     JRS16 RS-232-C P2                                1/5
       1 Device                    [Maintenance Cons]
       2 Speed(Baud rate)                    [9600 ]
       3 Parity bit                          [None ]
       4 Stop bit                            [1bit   ]
       5 Time out value(sec)                 [   0]

   [ TYPE ]              LIST    [CHOICE]
```

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 will be displayed.

```
SETUP Port Init
                                                 1/2
         Connector     Port      Comment
      1 JRS16 RS-232-C P2: [Maintenance Cons]
      2 JD17  RS-232-C P3: [KCL/CRT         ]

   [ 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.

8.3 FILES

A file is a unit of data storage in the memory of the robot controller. The following types of file are used mainly:

- Program file (*.TP)
- Default Logic File (*.DF)
- System file (*.SV) Used to store the settings of the system.
- I/O Config Data File (*.IO) Used to store the settings of Input/Output configuration.
- Date file (*.VR) Used to store data such as register data.

8.3.1 Program File

A program file contains a sequence of instructions for the robot. These instructions are called program instructions. Program instructions control robot operations, peripheral devices, and each application.
A program file is automatically stored in the memory of the controller. A directory of program files is displayed on the program selection screen.

> **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.
- Program name : Only the name of the program is displayed.

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 key which is displayed in the next page.

- DF_LOGI1.DF F2 key
- DF_LOGI2.DF F3 key
- DF_LOGI3.DF F4 key

8.3.3 System File/Application File

A system file/application file (*.SV) contains a system control program for operating the application tool software, or contains data used with the system. The following types of system file are used:

- SYSVARS.SV : Used to store the settings of the system variables relative to the, reference points, joint operating area and brake control.
- SYSFRAME.SV : User frame and tool frame
- SYSSERVO.SV : Used to store servo parameter data.
- SYSMAST.SV : Used to store mastering data.
- SYSMACRO.SV : Used to store the settings of the macro command.
- FRAMEVAR.VR : Used to store the settings of the reference position which is used at setting the frame, comments, etc. Frame data is stored to SYSFRAME.SV.

8.3.4 Data File

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.
  - STRREG.VR : Used to store the data of the string register.
  - PALREG.VR : Used to store the data of the palletizing register.
- 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 robot model.

8.3.5 ASCII File

An ASCII file (*.LS) is a file of ASCII format. To read an ASCII files, the optional function for ASCII upload is required. The contents of an ASCII file can, however, be displayed and printed using a personal computer.

8.4 SAVING FILES

The function of saving files stores the data which exists in the RAM memory in the controller to the external storage device. The following screens on the teach pendant can be used to save the files.

- Program selection screen: A specified program is saved to a storage device as program files.
- File screen: The specified program file, system file, etc can be saved to a storage device. The following files can be saved: When a batched save operation is executed, program files, system files, and application files can all be saved at the same time.
  - Program file
  - System file
  - Application file
  - Default logic file
- "5 SAVE" in the function menu: It is possible to preserve it on the storage device as program file and a system file, etc. of the program and the data, etc. displayed on the screen. The following files can be preserved:
  - Program file
  - System file
  - Data file
  - Application file
  - Default logic file
  - Application TP program file
  - Error log file
  - Diagnostic file
  - Vision data file
  - ASCII program file

8.4.1 Saving with Program Selection Screen

Program selection screen enables you to save the specified program as the program file.

Procedure 8-10 Saving program files in the program selection screen

Condition

- ■ The file input/output device is set correctly. (See Section 8.1.)

Step

1 Press [MENU] key to display the screen menu.
2 Select NEXT and then select "1 SELECT" on the next page, or press the "SELECT" key. The program selection screen will be displayed.

```
Select
                  1014788 bytes free       9/10
       No.   Program name            Comment
        1  -BCKEDT-          [                 ]
        2  GETDATA       MR  [Get PC Data      ]
        3  REQMENU       MR  [Request PC Menu  ]
        4  SENDDATA      MR  [Send PC Data     ]
        5  SENDEVNT      MR  [Send PC Event    ]
        6  SENDSYSV      MR  [Send PC Sysvar   ]
        7  SAMPLE1           [SAMPLE PROGRAM1  ]
        8  SAMPLE2           [SAMPLE PROGRAM2  ]
        9  SAMPLE3           [SAMPLE PROGRAM3  ]
       10  PROG1             [PROGRAM001       ]

    [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR ]  >
      COPY    DETAIL   LOAD   SAVE AS   PRINT   >
```

3 Press the [NEXT] key,>, and press F4, SAVE on the next page. The program save screen will be displayed.

```
SAVEAS
     From Path    :                        1/3
      MD:\
     From         :
      SAMPLE3.TP

     To  :        [MC:       ]
     To Directory :
      \
     To Filename  :
      SAMPLE3.TP

   DO_SAVE              [CHOICE]  CANCEL
```

4 Press F4, [CHOICE] to select the device to store the file. If you want to store the file to sub directory in the selected device, enter the name of sub directory to the item "To Directory". If you want to change the file name in the storage device, enter the name of a program to be saved. Then, press the [ENTER] key. The specified program will be saved to the device.

NOTE

When you specify the name of the sub directory, enter "\" at the end of the name of the sub directory. In case that the specified sub directory does not exist, when F1, DO SAVE is pressed, the alarm "File not found" occurs, and the file cannot be saved.

5 When a program having a same name as you want to save exists in the device, the following question will be displayed. If you agree to overwrite the file, press F4, YES.

```
     Overwrite?

                        YES      NO
```

8.4.2 Saving all the Program Files Using the File Screen

File screen enables you to save the program file or system file which is saved in the RAM memory in the device.
The following files can be saved by pressing F4, [BACKUP]:

Table 8.4.2 The saved files by each item in F4 BACKUP menu

Item in BACKUP menu	Saved files	Description
System files	System files (*.SV, *.VR)	Used to store the following files. - System variable file (SYSVARS.SV) - Servo parameter file (SYSSERVO.SV) - Mastering data file (SYSMAST.SV) - Macro data file (SYSMACRO.SV) - Frame setup file (FRAMEVAR.VR) - Frame data file (SYSFRAME.SV)
	Data files (*.IO, *.VR, *.DT)	Used to store the following files. - Register data file (*.VR) - I/O configuration data file (*.IO) - Robot setting data file (*.DT)
TP programs	Program files (*.TP)	Used to store all the programs file which has contents of programs.
	Default logic files (*.DF)	Used to store the settings of default logic instructions.
Application	Application files (*.SV)	Used to save the settings of an application. (Example: SYSSPOT.SV)
Applic. TP	Application TP program files (*.TP, *.DF, *.MN)	Used to store the program for the specific application.
Error log	Error log files (*.LS)	Used to store the alarm log in ASCII format.
Diagnostic	Diagnostic files (*.DG)	Used to store the specific diagnostic file in ASCII format.
Vision data	Vision data files (*.VD)	Used to store the vision data, setting of camera and camera calibration data and so on.
All of above	All of above files	Used to store all of above files. (The following files are not included.)
Maintenance	Diagnostic log files	Used to store the internal data while some problems occur. For detail about this function, refer to 9.15 DIAGNOSTIC LOG GETTING FUNCTION.
ASCII program	ASCII file (*.LS)	Used to store all programs in ASCII format.
Image	Image files (*.IMG)	Used to store the images of F-ROM and S-RAM memories of the controller. For detail, refer to 8.9 IMAGE BACKUP FUNCTION.

To interrupt the saving, press the [PREV] key while saving.

NOTE

At controlled start time, F4 is set to RESTOR instead of BACKUP. When RESTORE/BACKUP is selected from the function menu, BACKUP is displayed.

Procedure 8-11 Saving files in the file screen

Condition

■ The file input/output device is set correctly. (See Section 8.1.)

Step

1 Press [MENU] key to display the screen menu.
2 Select “7 FILE.” The file screen will be displayed.

```
FILE
     MC:\*.*                                1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Press DIR to generate directory

   [ TYPE ] [ DIR ]   LOAD   [BACKUP] [UTIL]  >
   DELETE    COPY   DISPLAY [INSTALL] [VIEW ] >
```

Saving program files

1 Press F4, [BACKUP], then select “TPE programs”.

```
FILE
     MC:\*.*                                1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Save MC:\SAMPLE1.TP?

              EXIT     ALL      YES      NO
```

- F2, EXIT Ends saving the 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.

2 Select the desired function key. The program files are saved in the device.

```
FILE
     MC:\*.*                                1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Saving MC:\SAMPLE1.TP, please wait...

              EXIT     ALL      YES      NO
```

3 When the file which has the same name as you specified already exists in the device, the following message will be displayed.

```
FILE
     MC:\*.*                                1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     MC:\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.

1 Press F4, SAVE and select System files. The following file will be displayed.

```
FILE
     MC:\*.*                                1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Save MC:\FRAMEVAR.VR?

            EXIT   ALL   YES   NO
```

- F2, EXIT Ends saving the system files.
- F3, ALL Saves all the system file and the data file.
- F4 YES Saves the specified file (system, data).
- F5, NO Does not save the specified file (system, data). After the file has been saved, the system asks whether the next system file or data file is to be saved.

2 Select the desired function key. The system files or the data files are saved in the device.

```
FILE
     MC:\*.*                                1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Backing up to disk: MC:\SYSVARS.SV

            EXIT   ALL   YES   NO
```

3 When the file having the same name as you want to save exists in the device, the following message will be displayed.

```
FILE
     MC:\*.*                                  1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     MC:\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.

Batched save

1 Press F4, [BACKUP], then select ALL of above.

> ⚠ **CAUTION**
> In case that the backup directory is the root directory of the storage device which format type is FAT16, more than 240 files cannot be saved. In this case, please create new subdirectory to backup files. Please go to step 4.

```
FILE
     MC:\*.*                                  1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Delete MC:\ before backup files?

                                 YES     NO
```

2 When F4, YES is selected at Step1, the following confirmation message will be displayed.

```
FILE
     MC:\*.*                              1/28
       1   *     *    (all files)
       2   *     KL   (all KAREL source)
       3   *     CF   (all command files)
       4   *     TX   (all text files)
       5   *     LS   (all KAREL listings)
       6   *     DT   (all KAREL data files)
       7   *     PC   (all KAREL p-code)
       8   *     TP   (all TP programs)
       9   *     MN   (all MN programs)
      10   *     VR   (all variable files)
      11   *     SV   (all system files)
     Delete MC:\ and backup files?

                                 YES      NO
```

- When F4, YES is selected, all the files in the external memory unit are erased, then all the data is saved. Processing is interrupted using the [PREV] key. An interrupt occurs once the current file has been processed.

> ⚠ **CAUTION**
> Before a batched save operation, all files in the external memory unit are erased. Before executing a batched save operation, check the files in the external memory unit.

3 When F5, NO is selected at Step1, the following confirmation message will be displayed.

```
FILE
     MC:\*.*                              1/28
       1   *     *    (all files)
       2   *     KL   (all KAREL source)
       3   *     CF   (all command files)
       4   *     TX   (all text files)
       5   *     LS   (all KAREL listings)
       6   *     DT   (all KAREL data files)
       7   *     PC   (all KAREL p-code)
       8   *     TP   (all TP programs)
       9   *     MN   (all MN programs)
      10   *     VR   (all variable files)
      11   *     SV   (all system files)
     No delete but backup all files to MC:\?

                                 YES      NO
```

- When F4, YES is selected, all the files in the external memory unit are kept and all the data is saved.

If the files can not be saved in the root directory of the storage device, save the file according to the following procedure.

4 The following screen will be displayed.

```
FILE
     MC:\*.*                                  1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Can't use root dir., create dir. Backupxx?

                         YES      NO
```

- When F4, YES is selected, the following screen will be displayed.

```
FILE
     MC:\*.*                                  1/28
       1    *    *     (all files)
       2    *    KL    (all KAREL source)
       3    *    CF    (all command files)
       4    *    TX    (all text files)
       5    *    LS    (all KAREL listings)
       6    *    DT    (all KAREL data files)
       7    *    PC    (all KAREL p-code)
       8    *    TP    (all TP programs)
       9    *    MN    (all MN programs)
      10    *    VR    (all variable files)
      11    *    SV    (all system files)
 No delete but backup all files to MC:\Backupxx\

                         YES      NO
```

5 When F4, YES is selected, all the data are save into the sub directory created by the operation described in step 4.

Maintenance data

This outputs the data saved by diagnostic log to an external device in case that it is difficult to do cycle power therefore image backup is not saved. Refer to “DIAGNOSTIC LOG” section of the chapter “UTILITY” to know diagnostic log.

1 Press F4, [BACKUP], then select maintenance data.

```
      Save maintenance data. OK?

                         YES      NO
```

2 When F4, YES is selected, the maintenance data is saved.
3 Following confirmation message will be displayed when maintenance data remains in the external device at step 1. When F4, YES is selected, maintenance data is overwritten to save.

```
      Data existed. Overwrite?

                         YES      NO
```

Maintenance data is saved into MNT_DATA\ under the current directory. It outputs the data saved by diagnostic log into FROM that is available for the investigation of some abnormal status of the robot controller to the external device.

8.4.3 Saving with a Function Menu

By selecting SAVE from a function menu, the data of a screen currently displayed can be saved into the device. The data of the following screens can be saved:

- Program edit screen Program file (*.TP)
- System variable screen System variable file (SYSVARS.SV)
- Mastering screen Mastering data file (SYSMAST.SV)
- Macro setting screen Macro data file (SYSMACRO.SV)
- Frame setup screen Frame setup data file (FRAMEVAR.VR)
- Register screen Register data file (NUMREG.VR)
- Position register screen Position register data file (POSREG.VR)
- String register screen String register data file (STRREG.VR)
- Pallet register screen Pallet register data file (PALREG.VR)
- I/O screen I/O configuration data screen (DIOCFGSV.IP)
- Edit screen for each default logic instruction. Each default logic instruction. (*.DF)
- Spot I/O setup screen or Weld sequence screen Spot weld setup data file (SYSSPOT.SV)

Procedure 8-12 Saving with a function menu

Condition

■ The file input/output device is set correctly. (See Section 8.1.)

Saving program files.

- Step

1 Display the program edit screen or the program selection screen.

```
Select
                 1014788 bytes free       9/10
      No.   Program name         Comment
       1  -BCKEDT-           [                 ]
       2  GETDATA        MR  [Get PC Data      ]
       3  REQMENU        MR  [Request PC Menu  ]
       4  SENDDATA       MR  [Send PC Data     ]
       5  SENDEVNT       MR  [Send PC Event    ]
       6  SENDSYSV       MR  [Send PC Sysvar   ]
       7  SAMPLE1            [SAMPLE PROGRAM1  ]
       8  SAMPLE2            [SAMPLE PROGRAM2  ]
       9  SAMPLE3            [SAMPLE PROGRAM3  ]
      10  PROG1              [PROGRAM001       ]

   [ TYPE ]  CREATE   DELETE  MONITOR  [ATTR ]  >
```

2 To display a function menu, press [FCTN] key.
3 Select “0 --NEXT--“, then select ”2 SAVE”. A selected program file is saved.
4 When the program having the same name as you want to save exists in the device, the file can not be saved.

```
Select
                  1014788 bytes free       9/10
      No.   Program name         Comment
       1  -BCKEDT-           [               ]
       2  GETDATA       MR  [Get PC Data      ]
       3  REQMENU       MR  [Request PC Menu  ]
       4  SENDDATA      MR  [Send PC Data     ]
       5  SENDEVNT      MR  [Send PC Event    ]
       6  SENDSYSV      MR  [Send PC Sysvar   ]
       7  SAMPLE1           [SAMPLE PROGRAM1  ]
       8  SAMPLE2           [SAMPLE PROGRAM2  ]
       9  SAMPLE3           [SAMPLE PROGRAM3  ]
      10  PROG1             [PROGRAM001       ]
     File already exists

   [ TYPE ]  CREATE   DELETE   MONITOR  [ATTR ]  >
```

5 When the device is filled with the files, exchange the device.

```
Select
                  1014788 bytes free       9/10
      No.   Program name         Comment
       1  -BCKEDT-           [       ]
       2  GETDATA       MR  [Get PC Data    ]
       3  REQMENU       MR  [Request PC Menu]
       4  SENDDATA      MR  [Send PC Data   ]
       5  SENDEVNT      MR  [Send PC Event  ]
       6  SENDSYSV      MR  [Send PC Sysvar ]
       7  SAMPLE1           [SAMPLE PROGRAM1]
       8  SAMPLE2           [SAMPLE PROGRAM2]
       9  SAMPLE3           [SAMPLE PROGRAM3]
      10  PROG1             [PROGRAM001     ]
     Disk is full

   [ TYPE ]  CREATE   DELETE   MONITOR  [ATTR ]  >
```

Saving other files.

- Step

1 Display the screen you want to save.

```
DATA Registers
                                          1/200
    R[   1:COUNTER1         ]=12
    R[   2:                 ]=0
    R[   3:                 ]=0
    R[   4:                 ]=0
    R[   5:                 ]=0
    R[   6:                 ]=0
    R[   7:                 ]=0
    R[   8:                 ]=0
    R[   9:                 ]=0
    R[  10:                 ]=0
    R[  11:                 ]=0
    Enter value

   [ TYPE ]
```

2 Display the function menu by pressing [FCTN] key.
3 Select “0 --NEXT--“, then select ”2 SAVE”.
The contents of the screen being displayed are saved.
4 When the file having a same name exists in the device, the file will be overwritten.
5 When the device is filled with the files, exchange the device.

```
FLPY-005  Disk is full                    10%
DATA Registers
                                     1/200
   R[   1:COUNTER1           ]=12
   R[   2:                   ]=0
   R[   3:                   ]=0
   R[   4:                   ]=0
   R[   5:                   ]=0
   R[   6:                   ]=0
   R[   7:                   ]=0
   R[   8:                   ]=0
   R[   9:                   ]=0
   R[  10:                   ]=0
   R[  11:                   ]=0
   Enter value

 [ TYPE ]
```

8.4.4 File Manipulation

On the file screen, files saved on a device can be listed and a file can be copied or deleted.

Procedure 8-13 File manipulation

Condition

- ■ The file input/output device is set correctly. (See Section 8.1.)

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select 7 FILE. The file screen will be displayed.

```
FILE
     MC:\*.*                              1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Press DIR to generate directory

 [ TYPE ] [ DIR ]   LOAD  [BACKUP] [UTIL]  >
   DELETE    COPY   DISPLAY [INSTALL] [VIEW ]  >
```

Displaying the list of files.

3 Press F2, [DIR].

```
FILE
     MC:\*.*                           Directory Subset 4
       1   Directory Subset 2  Directory Subset 3
  Directory Subset 1
  1 *.*
  2 *.KL
  3 *.CF
  4 *.TX
  5 *.LS
  6 *.DT                                          iles
  7 *.PC                                          page --
  8 -- next page --  page --      page --
                           (all variable files)
      11   *      SV   (all system files)
     Press DIR to generate directory

 [ TYPE ] [ DIR ]   LOAD   [BACKUP] [UTIL]   >
```

4 Select ”*.*”(all files). The list of the files being saved into the device will be displayed.

```
FILE
     MC:\*.*                              1/34
       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)
       9   *       CF   (all command files)
      10   *       TX   (all text files)
      11   *       LS   (all KAREL listings)

 [ TYPE ] [ DIR ]   LOAD   [BACKUP] [UTIL ]   >
```

Deleting files

5 Select the file you want to delete and press F1, DELETE.

```
FILE
     MC:\*.*                              3/34
       1   PRG1                      TP     384
       2   PRG2                      TP     384
       3   PRG3                      TP     768
       4   SYSVARS                   SV   25600
       5   SYSMACRO                  SV     324
       6   NUMREG                    VR     708
       7   DIOCFGSV                  IO     476
       8   *       *    (all files)
       9   *       KL   (all KAREL source)
      10   *       CF   (all command files)
      11   *       TX   (all text files)
     Delete MC:\PRG3.TP?

                              YES      NO     >
```

NOTE
Deleting a program from memory of the controller does not automatically delete the identical program from a device.

6 Press F4, YES. The file will be deleted.

```
FILE
     MC:\*.*                              3/34
       1   PRG1                        TP     384
       2   PRG2                        TP     384
       3   <Deleted>
       4   SYSVARS                     SV   25600
       5   SYSMACRO                    SV     324
       6   NUMREG                      VR     708
       7   DIOCFGSV                    IO     476
       8   *       *    (all files)
       9   *       KL   (all KAREL source)
      10   *       CF   (all command files)
      11   *       TX   (all text files)
     Deleted file MC:\PRG3.TP

   DELETE    COPY    DISPLAY  [INSTALL]  [VIEW ]   >
```

Copying files

7 Select the file you want to copy and press F2, COPY.

```
FILE COPY
     From Path    :                       1/3
      MC:\
     From         :
      PRG2.TP

     To  :              [***     ]
     To Directory :
      \
     To Filename  :
      PRG2.TP

   DO_COPY                       [CHOICE]  CANCEL
```

8 Move the cursor to “To :”, then press F4, [CHOICE] and select the device to copy the file from the displayed menu.

```
FILE COPY
     From Path    :                       1/3
   ------------------
                  1
   1 FROM Disk (FR:)
   2 Backup (FRA:)
   3 RAM Disk (RD:)       [***     ]
   4 Mem card (MC:)
   5 Mem Device (MD:)
   6 USB Disk (UD1:)
   7 USB on TP (UT1:)
   8
   ------------------

                                 [CHOICE]
```

9 If you want to copy the file into the sub directory of the selected device, move the cursor to “To Directory” and press F4, CHANGE, then enter the name of sub directory.

> **NOTE**
> When you specify the name of the sub directory, enter “\” at the end of the name of the sub directory. In case that the specified sub directory does not exist, when the copy is executed, the alarm “File not found” occurs, and the file cannot be copied.

```
FILE COPY
     From Path    :                       2/3
      MC:\
     From Filename         :
      PRG2.TP

     To Device    :               [MC:     ]
     To Directory :
      \
     To Filename  :              Alpha input  1
      PRG2.TP                      Upper Case
                                   Lower Case
                                   Punctuation
                                   Options

   ABCDEF  GHIJKL  MNOPQR  STUVWX  YZ_@*
```

10 Move the cursor to "To Filename" and F4, CHANGE, then enter the output file name.

```
FILE COPY
     From Path    :                       3/3
      MC:\
     From Filename         :
      PRG2.TP

     To Device    :               [MC:     ]
     To Directory :
      \
     To Filename  :              Alpha input 1
      PRG2.TP                      Upper Case
                                   Lower Case
                                   Punctuation
                                   Options

   ABCDEF  GHIJKL  MNOPQR  STUVWX  YZ_@*
```

11 When F1, DO_COPY is pressed, the file copy is executed. After the file copy is finished, the following message will be displayed.

```
FILE
     MC:\*.*                            3/34
       1   PRG1                       TP     384
       2   PRG2                       TP     384
       3   PRG3                       TP     768
       4   SYSVARS                    SV   25600
       5   SYSMACRO                   SV     324
       6   NUMREG                     VR     708
       7   DIOCFGSV                   IO     476
       8   *       *    (all files)
       9   *       KL   (all KAREL source)
      10   *       CF   (all command files)
      11   *       TX   (all text files)
     Copied to MC:\PRG3.TP

   DELETE   COPY   DISPLAY [INSTALL] [VIEW ]  >
```

Displaying the contents of files

12 It is possible to display the contents of the ASCII files. Select the file that you want to display the contents, then press F3, DISPLAY. The contents of the selected file are displayed. In the following example, the contents of ERROR.LS are displayed.

```
FILE Display
     MC:\*.*                              67/200
     ERROR.LS        Robot Name ROBOT 12- 1-24

     257” 12- 1-24    14:50” R E S E T
     258” 12- 1-24    14:50” R E S E T
     259” 12- 1-24    14:50” SYST-043 TP disable
     260” 12- 1-24    14:50” SRVO-223 DSP dry ru
     261” 12- 1-24    14:50” SYST-026 System nor
     262” 12- 1-24    14:50” HOST-178 Router Add
     263” 12- 1-24    14:50” HOST-108 Internet a
     264” 12- 1-24    14:50” FILE-069 USB hub In
     265” 12- 1-24    14:50” FILE-071 USB vend:
     Continue displaying?

                                  YES      NO
```

13 When F4, YES is pressed, the rest of the file is displayed. When F5, NO is pressed, display of the contents of the file is stopped and the previous screen is displayed. When the [PREV] key is pressed, display of the contents of the file is stopped and the previous screen is displayed, too.

14 If the file which cannot be displayed is selected and F3, DISPLAY is pressed, the following message is displayed and the contents of the file are not displayed.

```
FILE
     MC:\*.*                            3/34
       1   PRG1                     TP     384
       2   PRG2                     TP     384
       3   PRG3                     TP     768
       4   SYSVARS                  SV   25600
       5   SYSMACRO                 SV     324
       6   NUMREG                   VR     708
       7   DIOCFGSV                 IO     476
       8   *                        *(all files)
       9   *       KL   (all KAREL source)
      10   *       CF   (all command files)
      11   *       TX   (all text files)
     Cannot display file MC:\NUMREG.VR

     DELETE    COPY    DISPLAY [INSTALL] [VIEW ]  >
```

Changing the display

15 It is possible to change the display type of the file screen. Select the display type from, Normal, Name Only, Wide, by using F5, [VIEW] key.

(1) Normal : File name and file size are displayed.

```
FILE
     MC:\*.*                            3/34
       1   PRG1                     TP     384
       2   PRG2                     TP     384
       3   PRG3                     TP     768
       4   SYSVARS                  SV   25600
       5   SYSMACRO                 SV     324
       6   NUMREG                   VR     708
       7   DIOCFGSV                 IO     476
       8   *                        *(all files)
       9   *       KL   (all KAREL source)
      10   *       CF   (all command files)
      11   *       TX   (all text files)

     [ TYPE ] [ DIR ]   LOAD   [BACKUP] [UTIL ]  >
```

(2) Name Only : Only file name is displayed.

```
FILE
     MC:\*.*                                 3/34
       1   PRG1                            TP
       2   PRG2                            TP
       3   PRG3                            TP
       4   SYSVARS                         SV
       5   SYSMACRO                        SV
       6   NUMREG                          VR
       7   DIOCFGSV                        IO
       8   *       *    (all files)
       9   *       KL   (all KAREL source)
      10   *       CF   (all command files)
      11   *       TX   (all text files)

   [ TYPE ] [ DIR ]   LOAD   [BACKUP] [UTIL ]  >
```

(3) Wide : File name, file size, and file creation data are displayed. (The size of the character is reduced.)

```
FILE
  MC:\*.*                          3/34
   1      PRG1                           TP  384     21-JUL-2011 12:34
   2      PRG2                           TP  384     21-JUL-2011 12:34
   3      PRG3                           TP  384     21-JUL-2011 12:34
   4      SYSVARS                        SV  25600   21-JUL-2011 12:34
   5      SYSMACRO                       SV  324     21-JUL-2011 12:34
   6      NUMREG                         VR  708     21-JUL-2011 12:34
   7      DIOCFGSV                       IO  476     21-JUL-2011 12:34
   8      *         *     (all files)
   9      *         KL    (all KAREL source)
  10      *         CF    (all command files)
  11      *         TX    (all text files)
  12      *         LS    (all KAREL listings)
  13      *         DT    (all KAREL data files)
  14      *         PC    (all KAREL p-code)
  15      *         TP    (all TP programs)
  16      *         MN    (all MN programs)
  17      *         VR    (all variable files)
  18      *         SV    (all system files)
  19      *         IO    (I/O config data)
  20      *         DF    (all DEFAULT files)

  [ TYPE ] [ DIR ]   LOAD   [BACKUP] [UTIL ]  >
```

8.4.5 ASCII save

ASCII save function saves the program which is saved in binary (internal) format to the external memory device in ASCII format. This ASCII format is like the output of the printer.
The programs, which are saved to the storage device using this function, can be loaded to the personal computer and can be edited by it. Moreover, in order to load the program which is saved to the storage device in ASCII format into the robot controller, the software option is required.

The decimal place of the output position data is 3 in standard setting. The decimal place of the output position data can be changed from 0 to 8 by setting the system variable $LUPS_DIGIT.

File input/output device

The ASCII save function saves a file of ASCII format to a file input/output device selected according to Section 8.1.

Procedure 8-14 Execution of ASCII save function

Condition

■ The file input/output device is set correctly. (See Section 8.1.)

Step

1 Press [MENU] key to display the screen menu.

2 Select SELECT on the next page. The program directory screen will be displayed.

```
Select
                  1014788 bytes free      9/10
      No.   Program name          Comment
       1  -BCKEDT-          [                 ]
       2  GETDATA       MR  [Get PC Data      ]
       3  REQMENU       MR  [Request PC Menu  ]
       4  SENDDATA      MR  [Send PC Data     ]
       5  SENDEVNT      MR  [Send PC Event    ]
       6  SENDSYSV      MR  [Send PC Sysvar   ]
       7  SAMPLE1           [SAMPLE PROGRAM1  ]
       8  SAMPLE2           [SAMPLE PROGRAM2  ]
       9  SAMPLE3           [SAMPLE PROGRAM3  ]
      10  PROG1             [PROGRAM001       ]

   [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR ]  >
     COPY    DETAIL   LOAD   SAVE AS   PRINT   >
```

3 Press PRINT on the next page. The program print screen will be displayed .

```
         --- Print Teach Pendant Program  ---


      Program Name:
       SAMPLE3

                                        Alpha input 1
                                          Words
                                          Upper Case
                                          Lower Case
     Enter program name                   Options
      RSR     PNS     STYLE     JOB     TEST
```

4 Enter the name of the program to be saved with the ASCII save function, then press the [ENTER] key.

5 The specified program is saved with the ASCII save function. A file is saved with extension LS. In the same way, print data can be output as a file of ASCII format by print operation based on the function menu (Section 8.6).

8.5 LOADING FILES

Loading files is to load the files being saved in the file input/output device to the memory in the controller. The files can be loaded with the following screens on the teach pendant:

- Program selection screen -The specified program file is loaded from the device as the program.
- File screen -The specified program files and system files can be loaded. The following files can be loaded.
  - Program file (*.TP or *.MN)
  - Default logic instruction (*.DF)
  - System file (*.SV)
  - Application file (SYSSPOT.SV)
  - Data file (*.VR,*.IO)

NOTE

Selecting F4, [RESTOR] on the file screen in the controlled start (not controlled 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.

⚠ WARNING

Before a program set as a macro instruction is copied from a controller onto another controller, the macro setting screens of the two controllers should be compared. Be sure that the lists of the two controllers match. The program should be copied only when the lists match. Otherwise, an unpredictable result would occur that could injure personnel or damage equipment.

8.5.1 Loading Using the Program Selection Screen

In the program selection screen, the specified program file can be loaded from a device.

Procedure 8-15 Loading a program file using the program selection screen

Condition

- The file input/output device is set correctly. (See Section 8.1.)

Step

1 Press [MENU] key to display the screen menu.
2 Select ”0 -- NEXT --” and select ”1 SELECT” from the next page.
Program selection screen will be displayed.

```
Select
                  1014788 bytes free      9/10
      No.   Program name          Comment
       1  -BCKEDT-          [                 ]
       2  GETDATA       MR  [Get PC Data      ]
       3  REQMENU       MR  [Request PC Menu  ]
       4  SENDDATA      MR  [Send PC Data     ]
       5  SENDEVNT      MR  [Send PC Event    ]
       6  SENDSYSV      MR  [Send PC Sysvar   ]
       7  SAMPLE1           [SAMPLE PROGRAM1  ]
       8  SAMPLE2           [SAMPLE PROGRAM2  ]
       9  SAMPLE3           [SAMPLE PROGRAM3  ]
      10  PROG1             [PROGRAM001       ]

   [ TYPE ]  CREATE   DELETE   MONITOR  [ATTR ]  >
     COPY    DETAIL    LOAD    SAVE AS   PRINT   >
```

3 Press ”NEXT”,>, and press F3, LOAD, on the next page. Program load screen will be displayed.

```
--- Load Teach Pendant Program ---

Program name    :


                                  Alpha input 1
                                    Words
                                    Upper Case
                                    Lower Case
Enter program name                  Options
 RSR | PNS | STYLE | JOB | TEST
```

4 Enter the name of a program to be loaded, then press the [ENTER] key.

> **NOTE**
> Do not include a file extension in the program name.

A specified program is loaded from a device.

5 When the program having the same name as you want to load exists in the memory, the following message is displayed.

```
--- Load Teach Pendant Program ---

Program name    :
 PROG1



PROG1 already exists, select function
 |  | OVERWRITE |  | CANCEL
```

- OVERWRITE Loads the new file and overwrites it.

8.5.2 Loading a Specified Program File Using the File Screen

In the file screen, the specified file is loaded from the device to the memory.
The following files can be read:

- Program file (*.TP or *.MN) -Program file having contents of the program can be loaded.
- Default logic file (*.DF) -Default logic file having the settings of the default logic instruction can be loaded. The method of loading is the same as the program file.
- Data file (*.VR,*.IO) -The following data file can be loaded.
  - Register data file (NUMREG.VR)
  - Position register data file (POSREG.VR)
  - String register data file (STRREG.VR)
  - Palletizing register data file (PALREG.VR)

  - I/O config data file (DIOCFGSV.IO)
    (When DIOCFGSV.IO is loaded, cycle power or cold start is needed.)
- System file (*.SV) -The following system files can be loaded. However, system files can be loaded only at the controlled start. (See Subsection B.1.3, "Controlled start".)
  - System variable file (SYSVARS.SV)
  - Servo parameter file (SYSSERVO.SV)
  - Mastering data file (SYSMAST.SV)
  - Macro data file (SYSMACRO.SV)
  - Frame setup screen data file (FRAMEVAR.VR)
  - Frame data file (SYSFRAME.SV)

Procedure 8-16 Loading a program file using the file screen

Condition

■ The file input/output device is set correctly. (See Section 8.1.)

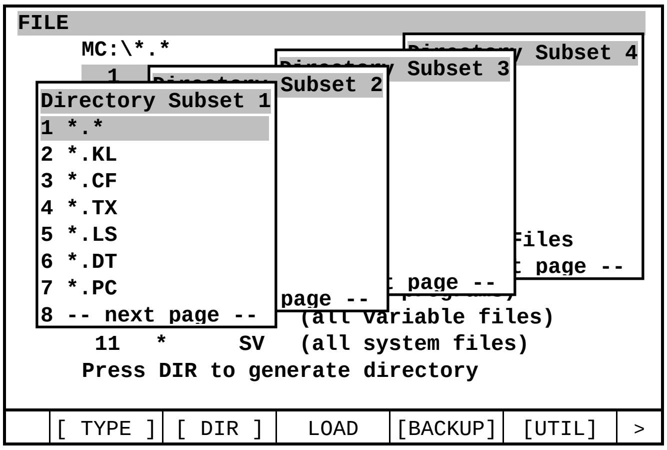
Step

1 Press [MENU] key to display the screen menu.
2 Select "7 FILE" to display the file screen.

```
FILE
     MC:\*.*                                1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Press DIR to generate directory

   [ TYPE ] [ DIR ]   LOAD   [BACKUP]  [UTIL]   >
   DELETE    COPY   DISPLAY [INSTALL] [VIEW ]   >
```

Loading a program file

3 Press F2, DIR.



4 Select "*.TP" (program file). The directory of program files stored on the device will be displayed.

```
FILE
     MC:\*.TP                             1/32
       1   PROGRAM1                  TP      768
       2   PROGRAM2                  TP      384
       3   TEST1                     TP     6016
       4   TEST2                     TP      704
       5   *        *     (all  files)
       6   *        KL    (all KAREL source)
       7   *        CF    (all command files)
       8   *        TX    (all text files)
       9   *        LS    (all KAREL listings)
      10   *        DT    (all KAREL data files)
      11   *        PC    (KAREL p-code)

   [ TYPE ] [ DIR ]   LOAD   [BACKUP]  [UTIL]  >
```

5 Move the cursor to the program file you want to load and press F3, LOAD.

```
FILE
     MC:\*.TP                             1/32
       1   PROGRAM1                  TP      768
       2   PROGRAM2                  TP      384
       3   TEST1                     TP     6016
       4   TEST2                     TP      704
       5   *        *     (all files)
       6   *        KL    (all KAREL source)
       7   *        CF    (all command files)
       8   *        TX    (all text files)
       9   *        LS    (all KAREL listings)
      10   *        DT    (all KAREL data files)
      11   *        PC    (all KAREL p-code)
      Loading MC:\PROGRAM1.TP, Prev to exit.

   [ TYPE ] [ DIR ]   LOAD   [BACKUP]  [UTIL]  >
```

Selected program is loaded from the device.

```
FILE
     MC:\*.TP                             1/32
       1   PROGRAM1                  TP      768
       2   PROGRAM2                  TP      384
       3   TEST1                     TP     6016
       4   TEST2                     TP      704
       5   *        *     (all files)
       6   *        KL    (all KAREL source)
       7   *        CF    (all command files)
       8   *        TX    (all text files)
       9   *        LS    (all KAREL listings)
      10   *        DT    (all KAREL data files)
      11   *        PC    (all KAREL p-code)
      Loaded MC:\PROGRAM1.TP

   [ TYPE ] [ DIR ]   LOAD   [BACKUP]  [UTIL]  >
```

6 If a program with the same name already exists in the RAM, the following indication will be provided:

```
FILE
     MC:\*.TP                              1/32
       1   PROGRAM1                    TP       768
       2   PROGRAM2                    TP       384
       3   TEST1                       TP      6016
       4   TEST2                       TP       704
       5   *        *     (all files)
       6   *        KL    (all KAREL source)
       7   *        CF    (all command files)
       8   *        TX    (all text files)
       9   *        LS    (all KAREL listings)
      10   *        DT    (all KAREL data files)
      11   *        PC    (all KAREL p-code)
      MC:\PROGRAM1.TP already exists

 |        |        |OVERWRITE|  SKIP  | CANCEL | >
```

- OVERWRITE Loads the new file and overwrites it.
- SKIP Skips to the next file.

7 If you want to load all program files, select ”*.TP” and press F3, LOAD. When the [PREV] key is pressed, the operation will be interrupted after the current a file is loaded.

Loading a data file

8 Press F2, DIR. The following menu will be displayed.

```
FILE
     MC:\*.*                          Directory Subset 4
                              Directory Subset 3
                   Directory Subset 2
 Directory Subset 1
 1 *.*
 2 *.KL
 3 *.CF
 4 *.TX
 5 *.LS                                        Files
 6 *.DT                                       t page --
 7 *.PC                                t page --
 8 -- next page --           page --
                              (all variable files)
      11   *      SV   (all system files)
     Press DIR to generate directory

 |[ TYPE ]|[ DIR ]|  LOAD  |[BACKUP]|[UTIL] | >
```

9 Select “*.VR” (variable data file). The directory of variable data files stored on the device is displayed.

```
FILE
     MC:\*.VR                              1/32
       1   FRAMEVAR                    VR      7262
       2   NUMREG                      VR       729
       3   STRREG                      VR       864
       4   POSREG                      VR      1190
       5   PALREG                      VR       232
       6   *        *     (all files)
       7   *        KL    (all KAREL source)
       8   *        CF    (all command files)
       9   *        TX    (all text files)
      10   *        LS    (all KAREL listings)
      11   *        DT    (all KAREL data files)

 |[ TYPE ]|[ DIR ]|  LOAD  |[BACKUP]|[UTIL] | >
```

10 Select a variable data file you want to load and press F3, LOAD.

```
FILE
     MC:\*.TP                           1/32
       1   FRAMEVAR                VR     7262
       2   NUMREG                  VR      729
       3   STRREG                  VR      864
       4   POSREG                  VR     1190
       5   PALREG                  VR      232
       6   *        *    (all files)
       7   *        KL   (all KAREL source)
       8   *        CF   (all command files)
       9   *        TX   (all text files)
      10   *        LS   (all KAREL listings)
      11   *        DT   (all KAREL data files)
     Loading MC:\FRAMEVAR.VR, Prev to exit.

   [ TYPE ] [ DIR ]   LOAD   [BACKUP] [UTIL]  >
```

The specified program is loaded from the device. Loaded data is set as the current data.

```
FILE
     MC:\*.TP                           1/32
       1   FRAMEVAR                VR     7262
       2   NUMREG                  VR      729
       3   STRREG                  VR      864
       4   POSREG                  VR     1190
       5   PALREG                  VR      232
       6   *        *    (all files)
       7   *        KL   (all KAREL source)
       8   *        CF   (all command files)
       9   *        TX   (all text files)
      10   *        LS   (all KAREL listings)
      11   *        DT   (all KAREL data files)
     Loaded MC:\FRAMEVAR.VR

   [ TYPE ] [ DIR ]   LOAD   [BACKUP] [UTIL]  >
```

11 If you want to load all the file which has the same extension, select ”*.VR”, ”*.IO”, etc and press F3, LOAD.

```
FILE
     MC:\*.*                            10/32
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Press DIR to generate directory

   [ TYPE ] [ DIR ]   LOAD   [BACKUP] [UTIL]  >
```

Loading system variable files

Condition

- Turn on the controller with [PREV] and the [NEXT] key pressed. Then select [3 Controlled start]. (See Subsection B.1.3, "Controlled start"). The following simplified system starts.

```
Tool Setup
                                          1/6
     1 F Number:                       F00000
     2 KAREL Prog in select menu:  NO
     3 Remote device:                  UserPanel
     4 Intrinsically safe TP:          NO

   [ TYPE ]
```

12 Press [MENU] key, then select "5 File." The file screen will be displayed.

```
FILE
     MC:\*.*                              1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Press DIR to generate directory

   [ TYPE ] [ DIR ]   LOAD  [RESTORE] [UTIL]  >
```

13 Press F2, [DIR] to display the submenu.

```
FILE
     MC:\*.*                     Directory Subset 4
       1               Directory Subset 3
                    Subset 2
  Directory Subset 1
  1 *.*
  2 *.KL
  3 *.CF
  4 *.TX
  5 *.LS                                   files
  6 *.DT                                   page --
  7 *.PC                       page --
  8 -- next page --  page --  riable files)
      11   *      SV   (all system files)
     Press DIR to generate directory

   [ TYPE ] [ DIR ]   LOAD  [RESTORE] [UTIL]  >
```

14 Select "*.SV" (system variable data file). The list of the system files which are saved in the device will be displayed.

```
FILE
     MC:\*.SV                          1/38
       1  SYSVARS              SV      77567
       2  SYSSERVO             SV       5731
       3  SYSMAST              SV       1609
       4  SYSMACRO             SV       2569
       5  SYSFRAME             SV        293
       6  SYSUIF               SV       2382
       7  MIXLOGIC             SV       1363
       8  CELLIO               SV       1429
       9  SYSFSAC              SV        380
      10  SYSHOST              SV       1060
      11  *        *    (all files)

   [ TYPE ] [ DIR ]   LOAD   [RESTORE] [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.

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.

```
FILE
     MC:\*.SV                          1/38
       1  SYSVARS              SV      77567
       2  SYSSERVO             SV       5731
       3  SYSMAST              SV       1609
       4  SYSMACRO             SV       2569
       5  SYSFRAME             SV        293
       6  SYSUIF               SV       2382
       7  MIXLOGIC             SV       1363
       8  CELLIO               SV       1429
       9  SYSFSAC              SV        380
      10  SYSHOST              SV       1060
      11  *        *    (all files)
     Convert? PREV to exit.

                               YES      NO     >
```

17 Select “1 START (COLD)” from the function menu. The system is cold started.

Batched read

Step

1 Select a file screen in the controlled start (not controlled start 2).
2 Select F4, [RESTOR].
3 A message asking the user for confirmation will be displayed on the prompt line.

```
FILE
     MC:\*.*                              1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Restore from Memory card(OVRWRT)?

                          YES      NO
```

4 Select F4, YES. Then, the read operation starts.
Processing is interrupted using the [PREV] key. An interrupt occurs once the current file has been processed.

8.6 PRINTING FILES

Printing files is outputting the contents of a program, a data file, the contents of system variables, and so on to a file I/O device selected as described in the file I/O device selection section (→ 8.1), as ASCII format files. The image being displayed on the teach pendant screen can also be output (print screen).
Printing files can be executed by the following screens.

- Program selection screen: Can print the program files.
- ”4 PRINT” on the second page of the FCTN menu: Can print the contents of the following screens:
  - Program edit screen: Program detail information and contents of program.
  - System variable screen: System variable data

Procedure 8-17 Printing files using program selection screen

Condition

■ The file I/O device is in an output enabled status.

Step

1 Press [MENU] key to display the screen menu.
2 Select “1 SELECT” on the next page.
The program selection screen will be displayed.

```
Select
                  1014788 bytes free      9/10
      No.   Program name          Comment
       1  -BCKEDT-           [                 ]
       2  GETDATA        MR  [Get PC Data      ]
       3  REQMENU        MR  [Request PC Menu  ]
       4  SENDDATA       MR  [Send PC Data     ]
       5  SENDEVNT       MR  [Send PC Event    ]
       6  SENDSYSV       MR  [Send PC Sysvar   ]
       7  SAMPLE1            [SAMPLE PROGRAM1  ]
       8  SAMPLE2            [SAMPLE PROGRAM2  ]
       9  SAMPLE3            [SAMPLE PROGRAM3  ]
      10  PROG1              [PROGRAM001       ]

   [ TYPE ]  CREATE   DELETE   MONITOR  [ATTR ]  >
    COPY     DETAIL   LOAD     SAVE AS   PRINT   >
```

3 Press F5, PRINT on the next page.
The program print screen will be displayed.

```
   --- Print Teach Pendant Program  ---

   Program Name:
    SAMPLE3

                                   Alpha input  1
                                     Words
                                     Upper Case
                                     Lower Case
   Enter program name                Options

 | RSR | PNS | STYLE | JOB | TEST |
```

4 Enter the name of a program file to be printed out, then press the [ENTER] key.
5 The specified program file is printed out. To stop printing, press the [PREV] key.

Procedure 8-18 Printing using the function menu

Program printing

Condition

- The program edit screen is displayed.

```
SAMPLE1
                                        1/6
      1: J  P[1] 100% FINE
      2: J  P[2] 70% CNT50
      3: L  P[3] 1000cm/min CNT30
      4: L  P[4] 500mm/sec FINE
      5: J  P[5] 100% FINE
   [End]

 | POINT |  |  |  | TOUCHUP | > |
```

Step

1 Press [FCTN] key to display the function menu.
2 Press 0 NEXT, and select 4 PRINT.
3 The currently displayed program is printed.
To interrupt printing, press the [PREV] key.

System variable printing

Condition

■ The system variable screen is displayed.

```
SYSTEM Variables
                                              1/638
      1 $AAVM              AAVM_T
      2 $ABSPOS_GRP        ABSPOS_GRP_T
      3 $ACC_MAXLMT        150
      4 $ACC_MINLMT        0
      5 $ACC_PRE_EXE       0
      6 $ALM_IF            ALM_IF_T
      7 $ANGTOL            [9] of REAL
      8 $APPLICATION       [9] of STRING[21]
      9 $AP_ACTIVE         6
     10 $AP_AUTOMODE       FALSE
     11 $AP_CHGAPONL       TRUE

   [ TYPE ]  DETAIL
```

Step

1. Press [FCTN] key to display the function menu.
2. Press 0 – NEXT –, then select 4 PRINT.
3. A list of system variables is printed.
4. To print only lower level system variables, for example, to print the system variables in $PARAM_GROUP, open the screen of the target level, and perform steps 1 and 2 above.

```
SYSTEM Variables
                                            332/638
    327 $OVRD_RATE         5
    328 $OVRD_SETUP        OVRD_SETUP_T
    329 $PADJ_SCHNUM       10
    330 $PALCFG            PALCFG_T
    331 $PARAM2_GRP        MRR2_GRP_T
    332 $PARAM_GROUP       MRR_GRP_T
    333 $PARAM_MENU        [21] of STRING[21]
    334 $PASSNAME          [10] of PASSNAME_T
    335 $PASSSUPER         PASSNAME_T
    336 $PASSWORD          PASSWORD_T
    337 $PAUSE_PROG        *uninit*

   [TYPE ]  DETAIL
```

```
SYSTEM Variables
    $PARAM_GROUP[1]                           1/236
      1 $BELT_ENABLE       FALSE
      2 $CART_ACCEL1       800
      3 $CART_ACCEL2       400
      4 $CIRC_RATE         1
      5 $CONTAXISNUM       0
      6 $EXP_ENBL          FALSE
      7 $JOINT_RATE        1
      8 $LINEAR_RATE       1
      9 $PATH_ACCEL1       800
     10 $PATH_ACCEL2       400
     11 $PROCESS_SPD       2000.0

   [ TYPE ]  DETAIL
```

Procedure 8-19 Printing the displayed screen using the function menu (print screen)

Condition

- The desired screen to be printed out is displayed.

Step

1. Press [FCTN] key to display the function menu.
2. Press 0 –NEXT–, then select 3 PRINT SCREEN.
3. The displayed screen is printed out. ”¥” is printed as the part of the reversed display on the teach pendant. In 7DC3 and later, image file is also printed out.
   To stop printing, press the [PREV] key.

Procedure 8-20 Printing the displayed screen using the TPSNAP.PC (print screen)

Condition

- The desired screen to be printed out is displayed.
- System software version is 7DF3 series.

Step

1. Execute the TPSNAP.PC karel program. For information on executing a karel program, please refer to “9.21.5 How to Run KAREL Program”.
2. The displayed screen is printed out as image file.

> **NOTE**
> Text file is not printed out by using the TPSNAP.PC.

8.7 SUBDIRECTORIES

Subdirectories can be used to organize programs or files on the controller. Subdirectory can be made into a subdirectory.

> **NOTE**
> Double-byte characters and following characters cannot be used in subdirectory name.
> “*****”, ”.”, ”|”, ”<”, ” >”, ”**/**”, ”**:**”, ”\”, ”**?**”, ” ”(Space)
> When you make a subdirectory by other of this controller, for example PC, the name of subdirectory may have invalid characters. Please be careful.

Procedure 8-21 Making Subdirectories

Condition

- The file input/output device is set correctly. (See Section 8.1.)

Step

1. Press [MENU] key to display the screen menu.
2. Select “7 FILE”. The file screen will be displayed as follows.

```
FILE
     MC:\*.*                             1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Press DIR to generate directory

   [ TYPE ] [ DIR ]   LOAD   [BACKUP]  [UTIL]  >
```

3 Press F5, [UTIL] and select Make DIR.

4 Input the subdirectory name followed from "Directory name:". Press the [ENTER] key to finish the input of subdirectory name.

```
FILE
     MC:\*.*                             1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP progAlpha input  1
       9   *      MN   (all MN prog  Words
      10   *      VR   (all variabl  Upper Case
      11   *      SV   (all system   Lower Case
     Directory name: RC11            Options

     RSR      PNS     STYLE     JOB     TEST
```

5 Subdirectory is created and current PATH is changed to this subdirectory automatically.

```
FILE
     MC:\RC11\*.*                        1/29
       1   ..     (Up one level)   <DIR>
       2   *      *    (all files)
       3   *      KL   (all KAREL source)
       4   *      CF   (all command files)
       5   *      TX   (all text files)
       6   *      LS   (all KAREL listings)
       7   *      DT   (all KAREL data files)
       8   *      PC   (all KAREL p-code)
       9   *      TP   (all TP programs)
      10   *      MN   (all MN programs)
      11   *      VR   (all variable files)
     Press DIR to generate directory

   [ TYPE ] [ DIR ]   LOAD   [BACKUP]  [UTIL]  >
```

Procedure 8-22 Using Subdirectories

Condition

- The file input/output device is set correctly. (See Section 8.1.)

Step

1 Press [MENU] key to display the screen menu.

2 Select “7 FILE”. The file screen will be displayed.

3 Press F2, [DIR], and select Directories or *.*. Directory entry is displayed with angle brackets.

```
FILE
     MC:\RC11\*.*                      1/30
       1  RC11        <DIR>
       2  RC12        <DIR>
       3  *      *    (all files)
       4  *      KL   (all KAREL source)
       5  *      CF   (all command files)
       6  *      TX   (all text files)
       7  *      LS   (all KAREL listings)
       8  *      DT   (all KAREL data files)
       9  *      PC   (all KAREL p-code)
      10  *      TP   (all TP programs)
      11  *      MN   (all variable files)
     Press DIR to generate directory

[ TYPE ] [ DIR ]  LOAD  [BACKUP] [UTIL] >
```

4 Move the cursor to a subdirectory entry with up or down arrow key then press the [ENTER] key to select this entry. Current PATH is changed to this subdirectory and the list of files included in this subdirectory is displayed automatically.

```
FILE
     MC:\RC11\*.*                      1/29
       1  ..     (Up one level)   <DIR>
       2  *      *    (all files)
       3  *      KL   (all KAREL source)
       4  *      CF   (all command files)
       5  *      TX   (all text files)
       6  *      LS   (all KAREL listings)
       7  *      DT   (all KAREL data files)
       8  *      PC   (all KAREL p-code)
       9  *      TP   (all TP programs)
      10  *      MN   (all MN programs)
      11  *      VR   (all variable files)
     Press DIR to generate directory

[ TYPE ] [ DIR ]  LOAD  [BACKUP] [UTIL] >
```

5 ".." in the first line indicates the parent directory. Move the cursor to ".. (Up one level)" and press ENTER then PATH is moved to the one step upper subdirectory. In the root directory, ".." is not displayed.

Procedure 8-23 Deleting Subdirectories

Condition

- The file input/output device is set correctly. (See Section 8.1.)

Step

1 Press [MENU] key to display the screen menu.
2 Select “7 FILE”. The file screen will be displayed.
3 Press F2, [DIR], and select Directories or *.*. Directory entry is displayed with angle brackets.
4 Move the cursor to the subdirectory that you want to delete. Press F1, DELETE in the next page of the function key.
5 The message "Do you want to delete ?" is displayed. Then Press F4, YES, then selected subdirectory will be deleted. If any file exists in the selected subdirectory then the subdirectory can not be deleted. After all files are deleted in the subdirectory, try to delete the subdirectory again.

8.8 AUTOMATIC BACKUP

8.8.1 Overview of Automatic Backup

- Automatic Backup function performs the transaction of “all backup” in File menu automatically at the following timing.
  - The specified time (Up to 5 settings per day)
  - The specified DI is turned on.
  - Start up of the controller. (Interval can be specified.)
- The memory card (MC:) and the automatic backup area (FRA:) of F-ROM in the controller can be specified as a backup copy destination. The FRA: is specified by default.
- Automatic Backup function can manage many versions of backup in one device. Even if you backup the wrong programs or settings, you can load the previous version of backup. The number of versions to keep can be set from 1 up to 99. (Default is 2.)
- A storage device to be used for automatic backup need be previously initialized for automatic backup. Automatic backup will not be performed for any external storage device that has not been initialized for automatic backup.
  Therefore, if an attempt is made to cause a backup copy to be automatically created on a memory card that has not be initialized for automatic backup, its content will not be lost.
  The FRA: need not be initialized, since it is previously initialized.
- If the controller is turned off during automatic backup, or automatic backup is stopped immaturely, the latest backup copy is automatically restored into the system. No incomplete backup file is left in the storage unit, and the latest backup file can be read at any time.

NOTE
This function automatically saves all files. If the storage device used for automatic backup becomes faulty, the data saved in it may not be read. In case such an unforeseen accident takes place, it is necessary to save backups to another storage device such as a memory card as well.

8.8.2 Usable Memory Cards

Refer to “8.1.1 Memory card” about the memory card that can be used in automatic backup function.

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 Mbytes memory card.

If a memory card other then those recommended is used, a normal operation is not guaranteed, and a bad influence may occur on the controller.

8.8.3 Setting of Automatic Backup

MENU → “7 FILE” → F1, [TYPE] → “Auto Backup”. The following menu will be displayed.

```
AUTO BACKUP
                                     1/15
     1 Automatic Backup:           ENABLE
     2 Device:              Backup (FRA:)
     3  Sub directory:          [        ]
        Status -  Ready for auto backup
  Backup Schedule ----------------------
     4 Backup Time 1:               12:00
     5 Backup Time 2:               23:30
     6 Backup Time 3:               **:**
     7 Backup Time 4:               **:**
     8 Backup Time 5:               **:**
     9 Backup at DI rising:       DI[    0]
    10 Backup at Power up:          ENABLE
    11   Interval:                  7 Day
  Status Output -------------------------
    12 Backup in progress:        DO[    0]
    13 Erro occurs at backup:     DO[    0]
  Version Management --------------------
    14 Maximum number of versions:    5
    15 Loadable version:    09/07/28 12:00

  [ TYPE ] INIT_DEV         ENABLE  DISABLE
```

Automatic Backup works only when it is ENABLE.

Device and subdirectory to save.
Default of device is “Backup(FRA:)”.
Default of subdirectory is blank.

Current status of the device is displayed. *

Set the timer to backup.
Up to 5 settings.
To clear setting, press F4, CLEAR.

When the specified DI is turned on, backup is performed. (If index is 0, it is disabled.)

If it is ENABLE, backup is performed at start up.
The interval can be set.

The specified DO is turned on when backup is performed, or when error occurs at backup.
(→8.8.4 Perform automatic backup)

Settings to manage versions of backup.
(→8.8.5 Version management)
(→8.8.6 Restore the backup)

<u>Power-on time backup</u>

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.

<u>Initializing of the storage device *</u>

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. (Procedure 8-2 Formatting file I/O device)
2 Press F2, INIT_DEV.
3 Message “Initialize the device for auto backup?” is displayed. Press F4, YES.
4 Message “Enter number of versions to keep:” is displayed. Please enter the number (1 to 99) of versions to keep. Pressing only the enter key sets the number of backup copy versions to 2.

INIT_DEV deletes all files in the device, and create the special files and directories.

> ⚠ **CAUTION**
> INIT_DEV does not format the device.
> Please format the device in file menu (F5, UTIL → “Format”)(Procedure 8-2 Formatting file I/O device).

> ⚠ **CAUTION**
> In case that the format type of the storage device is FAT16, up to 255 files can be saved to the route directory. Please pay attention to this point, when you select the storage device or the format type of the storage device. In case that you want to save more than 256 files, save the files to subdirectory or format the storage device by FAT32 format type.

8.8.4 Perform Automatic Backup

When the specified condition is satisfied, automatic backup is performed.

```
FILE-077 Auto backup start (FRA:¥)           10%
AUTO BACKUP
                                        1/15
     1 Automatic Backup:              ENABLE
     2 Device:                 Backup (FRA:)
     3  Sub directory:            [        ]
        Status -   Auto backup in progress
   Backup Schedule -----------------------
     4 Backup Time 1:                  12:00
     5 Backup Time 2:                  23:30
     6 Backup Time 3:                  **:**
     7 Backup Time 4:                  **:**
     8 Backup Time 5:                  **:**
     9 Backup at DI rising:        DI[    0]
 [ TYPE ] INIT_DEV          ENABLE  DISABLE
```

Informing automatic backup was performed.
( ) : Device to save.

Informing automatic backup is in progress.

- When automatic backup is performed, the message "FILE-077 Auto backup start (FRA:\)" is displayed in the upper screen. And the device name to save is displayed in the case arc.
- While automatic backup is performed, "Auto backup in progress" is displayed on the "Status" line.
- When automatic backup is completed, the message "FILE-078 Auto backup complete" is displayed in the upper screen.
- When automatic backup is completed, "Ready for auto backup" is displayed on the "Status" line.
- If the backup-in-progress signal is set, the specified signal becomes on while automatic backup is performed.

```
FILE-079  Error Auto backup(xxx.xx)          10%
FILE-055  MC not detected
AUTO BACKUP
                                        1/15
     1 Automatic Backup:              ENABLE
     2 Device:                 Backup (FRA:)
     3  Sub directory:            [        ]
        Status -   Device is not ready!!
   Backup Schedule -----------------------
     4 Backup Time 1:                  12:00
     5 Backup Time 2:                  23:30
     6 Backup Time 3:                  **:**
     7 Backup Time 4:                  **:**
     8 Backup Time 5:                  **:**
     9 Backup at DI rising:        DI[    0]
 [ TYPE ] INIT_DEV          ENABLE  DISABLE
```

Informing automatic backup was performed.
Showing the file name if automatic backup is failed to save a specific file.

Cause of the failure.

- The message "FILE-079 Error Auto backup (xxxx.xx)" will be displayed in the upper screen if backup failed, for example, because no memory card has been inserted. And if backup failed at a specific file, the file name was shown in the case arc. Then, the cause of failure is displayed under the message.

- In this case, the robot will not enter an alarm state.
- If a backup error signal is set, the specified signal becomes on while automatic backup is in progress. The set signal reset when the next automatic backup is performed.

> ⚠ **CAUTION**
> Please do not turn off the power of the controller while automatic backup function is performed. In case that the power of the controller is turned off while automatic backup function is performed, a normal operation is not guaranteed, and a bad influence may occur on the controller.

```
Alarm : Hist
                                           2/15
     1 FILE-079 Error Auto backup
     2 FILE-077 Auto backup start  (MC:¥)
     3 SRVO-012 Power failure recovery

 [ TYPE ] [ VIEW ]  ACTIVE   CLEAR   DETAIL
```

User can check the history of messages about automatic backup.

- These messages displayed in the upper screen when Auto backup started, finished, and was failed are overwritten by next warning or alarm messages. History of these messages about automatic backup can be checked in the alarm history screen.
  [MENU] → 4 ALARM → F3, HIST

```
                 FILE-077  Auto backup start (FRA:¥)
                 TEST001 LINE 0 T1 ABORTED WORLD      10%
TEST001
                                         4/4
      1: J  P[1] 100% FINE
      2: L  P[2] 2000mm/sec CNT100
      3: L  P[3] 2000mm/sec CNT100
    [End]
```

Automatic backup is in progress.

User can continue present work.

- Unlike the normal backup process, automatic backup run in the background. So if automatic backup is performed, user continues the current work without regard to the automatic backup process.
- While automatic backup is in progress, it is impossible to perform normal backup (ref.8.4.2) to same device. If the operation is executed, normal backup is cancelled with the message “Backup in progress” in the bottom left of the screen. And automatic backup is continued.
  However, differ in device, both of the two kinds of backup can be performed at the same time.
- Equally, while normal backup is in progress, it is impossible to perform automatic backup to same device. If the operation is executed, automatic backup is cancelled with the message “FILE-079 Error Auto backup (xxxx.xx)” and “FILE-081 Backup in progress” in the upper screen. And normal backup is continued.
  However, differ in device, both of the two kinds of backup can be performed at the same time.

8.8.5 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.

> **NOTE**
> You can’t select and delete backup versions. If you decrease the number of backup version, decrease the number of “maximum number of versions”. So the backup versions are deleted from older version until the number. .

In case that the device is FRA:, if the size of a free storage area in F-ROM in the controller becomes smaller than 1 Mbytes, 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.

> **NOTE**
> You can’t take backup data from FRA:. If you need to take backup data, you have to set device MC: or USB: to “2 Device” before executing automatic backup.

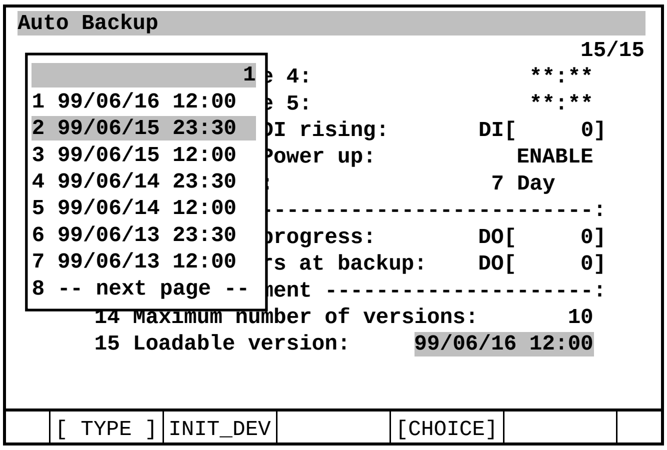
If it is impossible to hold a specified number of backup versions on a memory card because of an insufficient storage capacity, an error is detected during automatic backup execution. Specify an appropriate number of backup versions by assuming the storage capacity required to hold one backup version 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 cause an old backup version to be deleted, thus increasing a free area in the storage device.
Once a backup version is deleted by decreasing the value specified in “Maximum number of versions,” it cannot be restored by increasing the value.
Backup is stored in individual sub directories.
When automatic backup is performed, backup files are saved to the root directory, then these files are copied to the appropriate directory.
File menu can access the files only in root directory, so the latest version of backup can be loaded by file menu.
You can also load the older versions. (→ 8.8.6 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 controller is turned off during backup, or backup is stopped prematurely, all backup files created during the current backup session are deleted, and the last backup version selected is restored to the root directory.

8.8.6 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.



2 Please select the version to load, then the item “ Loadable version ” shows the time of the selected version. At this time, the files of the selected version of backup are copied to root directory.

3 You can load the files of the selected version in file menu.
When controlled start is performed, pressing all of above on the file menu of the controlled start menu enables all backup files to be read simultaneously.

8.9 IMAGE BACKUP FUNCTION

Overview

By using the image backup function, it is possible to create images of the F-ROM and S-RAM memories of the controller. These images are saved as several files to a selected storage unit.

The image backup function can be used from the FILE menu when the system is in cold start mode. After “Image backup” is selected from the menu, actual backups are made the next time the controller is turned on.

To restore backups, press and hold down the F1 and F2 keys and turn on the controller. A menu will be displayed from which previously saved images can be restored.

Storage units are a memory card (MC:), USB memory (UD1:, UT1:), and a PC server (TFTP:) connected via Ethernet. To use a PC server as a storage unit, it is necessary to set up the Ethernet function of the controller correctly start up the TFTP server function on the PC server.

It is possible to store the image file to the sub directory in the memory card (MC:) or USB memory (UD1:, UT1:). And it is possible to load the image file from the sub directory in the memory card (MC:) or USB memory (UD1:, UT1:), too.

Executing image backup

Procedure 8-24 Executing image backup

Step

1 Press [MENU] key
2 Select the FILE item.
3 Press F5, [UTIL] and select “Set device” in the menu. Then, select the device to backup the image file from “Mem card (MC:)”, “USB Disk (UD1:)”, “USB on TP (UT1:)” in the displayed menu.

NOTE
Tablet TP does not support getting IMG backup to UT1:.

4 Move the cursor to “ * * (all files)” and press the [ENTER] key.
5 Set the current directory to path what you want to backup by selecting directory from list of file and directory.

> ⚠ **CAUTION**
> Name of path should have only English or number. If path name have character other than English or number, character corruption occurs when restore image files.

6 Press the F4, [BACKUP] key. The menu below will be displayed.

```
          Backup 1
1 System files
2 TPE programs
3 Application
4 Applic.-TP
5 Error log
6 Diagnostic
7 Vision data
8 All of above
9 Maintenance data
0 -- NEXT --
```

```
          Backup 2
1 ASCII programs
2 Image backup
```

7 If the teach pendant is disabled when Image backup is selected, the prompt below will be displayed.

```
FILE
     MC:\*.*                          1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Enable TP for this operation

   [ TYPE ] [ DIR ]   LOAD   [BACKUP]  [UTIL]   >
```

Otherwise, the storage unit selection menu below will be displayed .

```
FILE
     MC:\*.*                             1/28
  Destination device  1 ll files)
  1 Current Directory   ll KAREL source)
  2 Ethernet(TFTP:)     ll command files)
                        ll text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Press DIR to generate directory

   [ TYPE ] [ DIR ]   LOAD   [BACKUP]  [UTIL]   >
```

8 If “1. Current Directory” is selected, the image backup files are saved to the currently selected directory. If the unit is MC: or UD1: or UT1: and it already contains *.IMG files, the confirmation message below will be displayed .

```
FILE
     MC:\*.*                              1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Remove existing IMG files ?

                              YES       NO
```

> ⚠ **CAUTION**
> If the storage unit is TFTP:, *.IMG files are always overwritten.

9 Select F4, YES, and a prompt will be displayed, requesting that the power be turned off and back on.

```
FILE
     MC:\*.*                              1/28
       1   *      *    (all files)
       2   *      KL   (all KAREL source)
       3   *      CF   (all command files)
       4   *      TX   (all text files)
       5   *      LS   (all KAREL listings)
       6   *      DT   (all KAREL data files)
       7   *      PC   (all KAREL p-code)
       8   *      TP   (all TP programs)
       9   *      MN   (all MN programs)
      10   *      VR   (all variable files)
      11   *      SV   (all system files)
     Cycle power?

                              OK        Cancel
```

10 Select F4, OK, and the power is automatically turned off and back on in R-30*i*B controller, so that the system restarts.
In R-30*i*B Mate, R-30*i*B Mate Plus, R-30*i*B Compact Plus or R-30*i*B Mini Plus controllers, please cycle power.

11 As soon as the power is turned on, image backup is automatically started. A screen such as that shown below will be displayed.

```
Writing MC:\FROM00.IMG
Writing MC:\FROM02.IMG

......
Writing MC:\SRAM02.IMG
DONE!
```

⚠ WARNING

1 During image backup, do not turn off the power.
2 If the storage unit is MC: or UD1: or UT1:, do not remove the external storage device during image backup.

12 After image backup is completed or if an error occurs, the controller completes an ordinary start sequence. When the start is completed, either of the screens below will be displayed.

```
FILE
      MC:\*.*                              1/28



      Image backup completed successfully.
|    |    |    | OK |    |
```

```
FILE
      MC:\*.*                              1/28



      Image backup failed.
|    |    |    | OK |    |
```

13 Press F4, OK, and the file screen automatically will be displayed.
14 If backup fails, alarm SYST-223 "Image backup failed (0x%x)" is posted to the alarm log. (0x%x) indicates the cause of the failure.

Restoring image backups

Procedure 8-25 Restoring image backups

Step

1. Press and hold down both F1 and F2 keys and turn on the robot. The image restore screen will be displayed.

```
***** Restore Controller Images *****
** Device selection menu ****
 1. Memory card(MC:)
 2. Ethernet(TFTP:)
 3. USB(UD1:)
 4. USB(UT1:)

  Select:
```

2. If an item other than 1 to 4 is selected, the controller starts an ordinary start sequence. If you want to select “1 Memory card(MC:)” or “3 USB(UD1:)” or “4 USB(UT1:), jump to the step 4. If the item “2 Ethernet (TFTP:)” is selected, the confirmation screen below will be displayed.

NOTE
Tablet TP does not support restoring IMG backup from UT1:.

```
*** BOOT MONITOR ***
Base version V8.10P/01    [Release2]
***** RESTORE Controller Images *****

Current memory module size
 FROM: 32Mb  SRAM: 3Mb

CAUTION: You SHOULD have image files
  from the same size of FROM/SRAM.
  If you don’t, this operation causes
  fatal damage to this controller.

Are you ready ? [Y=1/N=else]:
```

3. If 1 is entered, restoration starts. After completion, the controller starts an ordinary start sequence, using the restored images. If 2 is entered, image restoration is canceled.

4 In case that the item “1 Memory card(MC:)” is selected in the image restore screen, the following screen is displayed.

```
*** BOOT MONITOR ***
Base version V8.10P01      [Release2]
*** Directory selection menu ***
Current Directory:
MC:\
1.OK (Current Directory)
2.Directory1
3.Directory2
4.Directory3
5.Directory4
6.Directory5
7.Directory6
8.Directory7
Select[0.NEXT.-1.PREV]:
```

5 Up to 8 directories can be displayed at a time. (However, “OK(Current Directory)” is showed in the first line. And “.. (Up one level)” is showed in the second line if current directory is not root.) If you don’t find the destination directory, input 0 (NEXT) and press the [ENTER] key. Next directories will be displayed.

```
*** BOOT MONITOR ***
Base version V8.10P01      [Release2]
*** Directory selection menu ***
Current Directory:
MC:\
9.   Directory8
10.  Directory9
11.  Directory10
12.  Directory11


Select[0.NEXT,-1PREV]:
```

6 Select the destination directory. Current directory is changed to the selected sub directory and directory list including in the current directory are displayed. (If the directory which name has non alphanumeric character cannot be displayed in the list.)

```
*** BOOT MONITOR ***
Base version V8.10P01      [Release2]
*** Directory selection menu ***
Current Directory:
MC:\Directory10\
1.  OK (Current Directory)
2.  ..(Up one level)
3.  Directory10_1
4.  Directory10_2
5.  Directory10_3



Select[0.NEXT,-1PREV]:
```

7 Select “1 OK (Current Directory)”, if current directory is destination sub directory. If the number of the image files in the current directory corresponds with the memory module size of the controller, the following screen is displayed.

```
*** BOOT MONITOR ***
Base version V8.10P01     [Release2]
***** RESTORE Controller Images *****

Current memory module size
 FROM: xxMb  SRAM: xMb

Image files are detected in
MC:\Directory10\
 FROM: xxfiles  SRAM: xfiles
Correspond with module size.

Restore image files ? [Y=1/N=else]:
```

8 Select 1, if you are sure for selected path. Image files will be restored from selected path with following display.

```
Reading FROM00.IMG … Done
Reading FROM01.IMG … Done
Reading FROM02.IMG … Done
Reading FROM03.IMG … Done
Reading FROM04.IMG … Done
Reading FROM05.IMG … Done
Reading FROM06.IMG … Done
Reading FROM07.IMG … Done
Reading FROM08.IMG … Done
Reading FROM09.IMG … Done
Reading FROM10.IMG … Done
Reading FROM11.IMG … Done
```

8.10 ASCII PROGRAM LOADER FUNCTION

8.10.1 Overview

When robots are equipped with the ASCII Program Loader option, you can load teach pendant programs that are saved as ASCII files (.LS) directly onto the robot. ASCII teach pendant programs can be stored offline and manipulated without the specialized software required for handling binary programs. You can use general-purpose text editors such as those used for email to do the following.

This is optional function. To use this function, system must be installed “Ascii upload function (R796)”.

- Archive programs offline for code reuse in the same or different robots.
- Examine human-readable hardcopy programs away from a robot.
- Edit programs away from a robot.
- Generate programs using standard text-based scripting tools sold on the open market.

Robots equipped with the ASCII Program Loader option can read text-based, programs that were printed by the same or a compatibly configured robot.

The ASCII Program Loader expects the ASCII file to be of the format described in Section 8.10.4. This format is the same as that printed out by the controller. This function expects that you edit a part of TP program with this function. You should conduct the following process. Output TP programs -> Edit TP programs with text editor -> Load edited TP programs.

> ⚠ **CAUTION**
> If data in the ASCII file is incorrect or created without taking into account the robot and its location in the workcell, the ASCII Program Loader might produce a teach pendant program which, when loaded on a robot, could cause unexpected motion. This could result in damage to equipment and injury to personnel.
>
> Always step through the teach pendant program on the robot with motion locked, then run through the program with low speed and your hand on the teach pendant EMERGENCY STOP button.

8.10.2 Output an ASCII Teach Pendant Program

To output an Ascii Teach Pendant program, execute procedure 8-26.

Procedure 8-26 Output an ASCII Teach Pendant Program

Condition

- An external memory device must be connected properly.

Steps

1. Press the [MENU] key.
2. Select FILE.
3. Set the current directory to the external memory device that you want to put an Ascii Teach Pendant Program. (To put on a subdirectory, set the current directory to the subdirectory).
4. Press [SELECT] key to display select screen.as follows.

```
Select                                          i
        682368 bytes free                    3/18
     No.   Program name               Comment
       1 -BCKEDT-                [              ]
       2 CLSKP_SP                [              ]
       3 CLSKP_EX                [              ]
       4 CLSKP_G1                [              ]
       5 CLSKP_G2                [              ]
       6 CLSKP_G3                [              ]
       7 CLSKP_G4                [              ]
       8 CLSKP_G5                [              ]
       9 CLSKP_G6                [              ]
      10 CLSKP_G7                [              ]

   [TYPE]   CREATE   DELETE   MONITOR   [ ATTR]   >
```

5 Press the [NEXT] key to display next function keys as follows.

```
Select                                           i
            682368 bytes free               3/18
        No.   Program name             Comment
          1 -BCKEDT-               [              ]
          2 CLSKP_SP               [              ]
          3 CLSKP_EX               [              ]
          4 CLSKP_G1               [              ]
          5 CLSKP_G2               [              ]
          6 CLSKP_G3               [              ]
          7 CLSKP_G4               [              ]
          8 CLSKP_G5               [              ]
          9 CLSKP_G6               [              ]
         10 CLSKP_G7               [              ]

   | COPY | DETAIL | LOAD | SAVE AS | PRINT | >
```

6 Suit the key cursor to the target program and press F5[PRINT] key and the following screen will be displayed.

```
      -- Print Teach Pendant Program  ---

  Program Name:
   CLSKP_EX

                                           Alpha input 1
                                             Words
                                             Upper Case
                                             Lower Case
  Enter program name                         Options

   | RSR | PNS | STYLE | JOB | TEST | >
```

7 The program name that you selected at step 6 has been inputted already, so you press the [ENTER] key, then the Ascii program is output to the current directory that you selected at step 3. To change the target, change program name and press the [ENTER] key.

8.10.3 Loading an ASCII Teach Pendant Program from the Teach Pendant

Use procedure 8-27 when you load an ASCII teach pendant program.

> **NOTE**
> The programs you are loading must not be SELECTED or the load will fail with the message “MEMO-015 program already exists”.

Procedure 8-27 Loading an ASCII Teach Pendant Program

Condition

- The memory card must be connected properly.
- The program you want is on the memory card.
- The program features are compatible with controller’s configuration.

Steps

1 Press [MENU] key
2 Select FILE.
3 Set the default device to the memory card.

4 Select the desired filename.
   a. Press F2, [DIR] and move the cursor to *.LS and press the [ENTER] key. The following screen will be displayed.

```
FILE
     MC:\*.*                          1/29
       1   TESTSPOT              LS   16541
       2   *      *    (all files)
       3   *      KL   (all KAREL source)
       4   *      CF   (all command files)
       5   *      TX   (all text files)
       6   *      LS   (all KAREL listings)
       7   *      DT   (all KAREL data files)
       8   *      PC   (all KAREL p-code)
       9   *      TP   (all TP programs)
      10   *      MN   (all MN programs)
      11   *      VR   (all variable files)

   [ TYPE ] [ DIR ]   LOAD   [BACKUP] [UTIL]  >
```

5 Load the program.
   a. Move the cursor to the desired program and press F3, LOAD.
   b. Press YES. The following screen will be displayed.

```
FILE
     MC:\*.*                          1/29
       1   TESTSPOT              LS   16541
       2   *      *    (all files)
       3   *      KL   (all KAREL source)
       4   *      CF   (all command files)
       5   *      TX   (all text files)
       6   *      LS   (all KAREL listings)
       7   *      DT   (all KAREL data files)
       8   *      PC   (all KAREL p-code)
       9   *      TP   (all TP programs)
      10   *      MN   (all MN programs)
      11   *      VR   (all variable files)
     Loaded MC:\TESTSPOT.LS
   [ TYPE ] [ DIR ]   LOAD   [BACKUP] [UTIL]  >
```

NOTE
The TESTSPOT.TP file will be displayed in the SELECT menu.

NOTE
If there are errors, you will not be able to edit the ASCII program file on the controller. You will need to edit the ASCII program file with a text editor on another computer.

8.10.4 Viewing ASCII Program Loading Errors

When ASCII Program Loader detects a syntax error in the source file, it posts warning alarms indicating the location of the errors and terminates the load. The alarms can be viewed using the teach pendant ALARM HISTORY screen. The Alarm History screen allows you to display the items in Table 8.10.4 (a) and perform the operation in Table 8.10.4 (b). Refer to the Procedure 8-28 to display the ALARM HISTORY screen.

Table 8.10.4 (a) Viewing ASCII program loading error screen items

ITEM	DESCRIPTION
ASBN Alarms	The facility code ASBN alarms are generated by the ASCII Program Loader option.

Table 8.10.4 (b) Viewing ASCII program loading error screen operations

ITEM	DESCRIPTION
[TYPE]	Press this key to access various application-specific options.
HIST	Press this key to enter the alarm history screen.
ACTIVE	Press this key to access the active alarms.
CLEAR	Press this key to clear an ASCII program loading error.
DETAIL	Press this key to access detailed information on a particular ASCII Program loading error.

Procedure 8-28 Viewing ASCII Program loading Errors

Conditions

- .LS file was loaded and it failed.

Steps

1 Press [MENU] key and select ALARM. The following screen will be displayed.

```
Alarm : Active



        There are no active alarms.

      Press F3(HIST) to enter alarm





    [ TYPE ][ VIEW ]   HIST    RES_1CH
```

2 Press F3, HIST and move your cursor to ASBN-008 alarm. The following screen will be displayed.

```
Alarm : Hist

      1 ASBN-090 Undefined macro
      2 ASBN-090 Undefined macro
      3 ASBN-008 file ‘MD:MENUTEST.LS’
      4 ASBN-009 on line 22, column 13

    [ TYPE ][ VIEW ]  ACTIVE   CLEAR   DETAIL
```

NOTE

One or two ASBN-alarms followed by ASBN-008 will be displayed. The alarm will give you the file name and there will be one or more ASBN-009 alarms showing the line and column.

3. Move the cursor to ASBN 009 and press F5, HELP to view the DETAIL Alarm screen. The error shows the line and column after the offending word. The cause code shows an ASBN error indicating what is wrong.

```
Alarm : Hist
    DETAIL Alarm
    ASBN-009 on line 22, column 13
    ASBN-092 Undefined instruction
    WARN      09-01-23      12:34
    Alarm : Hist
      1 ASBN-090 Undefined macro
      2 ASBN-090 Undefined macro
      3 ASBN-008 file  'MD:MENUTEST.LS'
      4 ASBN-009 on line 22, column 13

   [ TYPE ][ VIEW ] ACTIVE  CLEAR  DETAIL
```

NOTE
If there are errors, you will not be able to edit the ASCII program file on the controller. You will need to edit the ASCII program name with a text editor on another computer.

8.10.5 Example ASCII File

This section details the syntax of an ASCII teach pendant program.

NOTE
You can transform the target TP program to the Ascii TP program and output it to the external memory device. Please make a target program by editing with text editor. (8.10.2 Output an Ascii Teach Pendant Program)

ASCII file contains examples of the following possible sections of an ASCII file.

- /PROG ···Program name
- /ATTR ···Program attribute data
- /APPL ···Tool application data
- /MN ···Teach Pendant instructions
- /POS ···Position data
- /END ···End of file

/PROG Assigns program name. You can assign program name consisting of one to 36 characters. To assign subtype, add a space and subtype name. Please refer to the following table in regard to subtype.

Table 8.10.5 (a) Assigning subtype

Subtype	Description
(blank)	Not assign subtype. This presents standard Teach Pendant Program
Macro	Assigns subtype Macro(MR). This presents a program for executing macro instruction.
Cond	Assigns subtype Condition(CH). This represents a condition program with the state monitoring function.
Job	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. This program attribute can be selected when application software is specific (such as Dispense) or when system variable $JOBPROC_ENB is set to 1.
Process	This represents a subprogram that is called by a job program for execution of a particular job. This program attribute can be selected when application software is specific (such as Dispense) or when system variable $JOBPROC_ENB is set to 1.

Ex. Assigns subtype Macro to TEST_PROG.TP

```
/PROG  TEST_PROG Macro
```

/ATTR Program attribute data.

This data contains of Owner, Comment, Program size, Created date, File name, Version, Line number, Memory size, File protect, TCD, Default group, and Control code. This data is for displaying attribute of TP program, so not assume that you edit this data and load. Please do not change this data in this item. But it is possible to edit only comment within 16 characters.

/APPL Specify application data that robot application tool software needs

- Handling

  No need for application data.

  However if multi application tools are installed to one robot controller, specify like the following to enable handling function.

  Ex: R-J3*i*B,R-30*i*A,R-30*i*B SpotTool+ Handling plug-in is ENABLE.

```
/APPL
  HANDLING : TRUE ;
```

- Palletizing

  Specify palletizing number, initial data, stacking pattern, path pattern conditions and path pattern when you want to use palletizing instruction in TP program.

  a Palletizing number

  Describe the palletizing number according to the following form.

```
PALLETIZING-[1];
```

  b Initial data

  This is data that specifies how palletizing is performed. This is different depending on the kind of palletizing. The initial data contains the following items:

Table 8.10.5 (b) Palletizing setting

Item	Description
PALLETIZING TYPE, NUMBER	This is palletizing type (Either B, BX, E or Ex) and palletizing number (1-16). The form is "PALLETIZING TYPE_NUMBER".
TYPE	This is stacking (PALLET) or unstacking (DEPALLET). If you select PALLET, palletizing register is incremented by the palletizing end instruction. If you select DEPALLET, palletizing register is decremented by the palletizing end instruction.
INCR	This is a value to increment or decrement the palletizing register by the palletizing end instruction.
PAL REG	This is a palletizing register to use in palletizing instruction.
ORDER	This is the stacking (unstacking) order of row, column, and layer. R: Row, C: Column, L: Layer

Item	Description
ROWS COLUMNS LAYERS (Arrangement mode) (Posture control) (Layer pattern count)	Those are the numbers of rows, columns, and layers for a stacking pattern. 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). Control the posture at rows, columns, and layers for a stacking pattern. Select E or EX. How workpieces are stacked can be specified for each layer (only for palletizing E or EX). 1 to 16.
APPR	Number of approach points in a path pattern. 0 to 8.
RTRT	Number of retraction points in a path pattern. 0 to 8.
PATTERN	Number of path patterns (only for palletizing BX or EX). 1 to 16.

Those data are described by the form indicated in the following example.

```
CONFIG:
    PALLETIZING-BX_1 "" ;
    TYPE =  PALLET ;
    INCR =   1 ;
    PAL REG = 1 ;
    ORDER = RCL ;
    ROWS = [5] ;
    COLUMNS = [4] ;
    LAYERS = [3] ;
    AUXILIARY POS = [NO ] ;
    APPR = 2 ;
    RTRT = 2 ;
    PATTERN = 3 ;
```

a Stacking pattern

Describe representative stack points of a stacking pattern according to stacking pattern specified by initial data. The form is the following. Stacking pattern that should be described is different depending on the content of the specified initial data. Refer to Chapter 10. PALLETIZING FUNCTION in this manual for this details.

```
BOTTOM:
    P[1,1,1]{
      GP1:
        UF : 0, UT : 1,    CONFIG : 'N U T, 0, 0, 0',
        X = 1600.000  mm,  Y =    0.000  mm,  Z = 1000.000  mm,
        W = -180.000 deg,  P =    -.000 deg,  R =    0.000 deg
    };
    P[5,1,1]{
      GP1:
        UF : 0, UT : 1,    CONFIG : 'N U T, 0, 0, 0',
        X = 1600.000  mm,  Y =    0.000  mm,  Z = 1000.000  mm,
        W = -180.000 deg,  P =    -.000 deg,  R =    0.000 deg
    };
    P[1,4,1]{
      GP1:
        UF : 0, UT : 1,    CONFIG : 'N U T, 0, 0, 0',
        X = 1600.000  mm,  Y =    0.000  mm,  Z = 1000.000  mm,
        W = -180.000 deg,  P =    -.000 deg,  R =    0.000 deg
    };
    P[1,1,3]{
      GP1:
```

```
            UF : 0, UT : 1,    CONFIG : 'N U T, 0, 0, 0',
            X = 1600.000  mm,  Y =    0.000  mm,  Z = 1000.000  mm,
            W = -180.000 deg,  P =    -.000 deg,  R =    0.000 deg
    };
```

b Path pattern conditions

For palletizing BX and EX, multiple path patterns can be set independently for stack points. This creates an association between each stack points and each path patterns in advance. The form is the following. Refer to Chapter 10. PALLETIZING FUNCTION in this manual for this details.

```
PATTERN:
    PTN [1] = [1, *, *] ;
    PTN [2] = [2, *, *] ;
    PTN [3] = [3, *, *] ;
```

c Path pattern

Describe path points according to path pattern specified by initial data. The form is the following. Path pattern that should be described is different depending on the content of the specified initial data. Refer to Chapter 10. PALLETIZING FUNCTION in this manual for this details.

```
ROUTE:
    ROUTE [1]{
      P[A_ 2]{
        GP1:
            UF : 0, UT : 1,    CONFIG : 'N U T, 0, 0, 0',
            X = 1600.000  mm,  Y =    0.000  mm,  Z = 1000.000  mm,
            W = -180.000 deg,  P =    -.000 deg,  R =    0.000 deg
      };
      P[A_ 1]{
        GP1:
            UF : 0, UT : 1,    CONFIG : 'N U T, 0, 0, 0',
            X = 1600.000  mm,  Y =    0.000  mm,  Z = 1000.000  mm,
            W = -180.000 deg,  P =    -.000 deg,  R =    0.000 deg
      };
      P[BTM]{
        GP1:
            UF : 0, UT : 1,    CONFIG : 'N U T, 0, 0, 0',
            X = 1600.000  mm,  Y =    0.000  mm,  Z = 1000.000  mm,
            W = -180.000 deg,  P =    -.000 deg,  R =    0.000 deg
      };
      P[R_1]{
        GP1:
            UF : 0, UT : 1,    CONFIG : 'N U T, 0, 0, 0',
            X = 1600.000  mm,  Y =    0.000  mm,  Z = 1000.000  mm,
            W = -180.000 deg,  P =    -.000 deg,  R =    0.000 deg
      };
      P[R_2]{
        GP1:
            UF : 0, UT : 1,    CONFIG : 'N U T, 0, 0, 0',
            X = 1600.000  mm,  Y =    0.000  mm,  Z = 1000.000  mm,
            W = -180.000 deg,  P =    -.000 deg,  R =    0.000 deg
      };
    };
```

- SPOT
  R-J3*i*B, R-30*i*A,R-30*i*B SpotTool+
  Specify spot welding function enable/disable. And specify spot welding equipment number used in TP programs.

```
/APPL
  SPOT : TRUE ;
  SPOT Welding Equipment Number : 1 ;
```

  R-J3*i*B SpotTool
  No need for application data.

- **ARC**
  Specify arc welding function enable/disable. And specify arc welding equipment number used in TP program.

```
/APPL
  ARC : TRUE ;
  ARC Welding Equipment : 1,*,*,*,*;
```

- **DISPENSE**
  R-30*i*B SpotTool+
  Specify dispense function enable/disable.

```
/APPL
  DISPENSE : TRUE ;
```

  R-30*i*A DispenseTool
  As for dispensing program, specify last cycle time, default user flame (used in TP program), default tool flame (used in TP program), and equipment number (used in TP program).

```
/APPL
DISPENSE_PROCESS;
   CYCLE_TIME              :0;
   LAST_CYCLE_TIME         :0;
   GUN_ON_TIME             :0;
   LAST_GUN_ON_TIME        :0;
   DEFAULT_USER_FRAME      :1;
   DEFAULT_TOOL_FRAME      :1;
   EQUIPMENT_NUMBER        :1;
```

  This setting is unnecessary for non-dispensing program.

  R-J3*i*B, R-30*i*A, R-30*i*B SpotTool+ with Dispense plug-in enabled

```
/APPL
  DISPENSE : TRUE ;
```

  R-J3*i*B SealingTool
  No application data description is needed.

/MN Write TP program in itself. (Motion instruction, Logic instruction, I/O instruction and so on)

/POS Specify Position data. (Joint positions or Cartesian positons for each group)

/END Presents End of file.

See an example of ASCII file in the following.

```
/PROG  TEST1
/ATTR
OWNER          = MNEDITOR;
COMMENT        = "";
PROG_SIZE      = 598;
CREATE         = DATE xx-05-01  TIME 22:23:54;
MODIFIED       = DATE xx-05-01  TIME 22:24:02;
FILE_NAME      = ;
VERSION        = 0;
LINE_COUNT     = 2;
MEMORY_SIZE    = 958;
PROTECT        = READ_WRITE;
TCD: STACK_SIZE          = 0,
     TASK_PRIORITY       = 50,
     TIME_SLICE          = 0,
     BUSY_LAMP_OFF       = 0,
     ABORT_REQUEST       = 0,
     PAUSE_REQUEST       = 0;
DEFAULT_GROUP  = 1,*,*,*,*,*,*,*;
CONTROL_CODE   = 00000000 00000000;
/APPL
  CYCLE_REFERENCE = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
  CYCLE_TARGET =    0.00 ;
/MN
   1:J P[1] 100% FINE     ;
   2:J P[2] 100% FINE     ;
/POS
P[1]{
   GP1:
        UF : 0, UT : 1,              CONFIG : 'F U T, 0, 0, 0',
        X =   1584.56   mm, Y =   299.91   mm,   Z =    300.3 mm,
        W =   -120.000 deg, P =      0.000 deg,  R =   -115.21 deg
};
P[2]{
   GP1:
        UF : 0, UT : 1,              CONFIG : 'F U T, 0, 0, 0',
        X =   1584.56   mm, Y =   299.91   mm,   Z =    300.3 mm,
        W =   -120.000 deg, P =      0.000 deg,  R =   -115.21 deg
};
/END
```

Program Name

Program Attribute Data

Application Data

Instructions

Start of Position Data

8.11 FILE MEMORY

You can check the amount of memory you are using in the file system using the File Memory screen. This will show the available memory in Kbytes (KB) for “FR:”, “RD:”, “MC:”, and “UD1:”.Use Procedure 8-29 to check file memory.

Procedure 8-29 Checking File Memory

Steps

1 Press [MENU] key
2 Select FILE.
3 Press F1, [TYPE].
4 Select File Memory. The following screen will be displayed.

```
File Memory


               Total(KB) Free(KB)  Type/Status
     Device -------------------------------
       FR:       31687     18620   FAT16
       MC:      499672    239860   FAT16
       RD:          59        53   FAT16
      UD1:      499672    301240   None

   [ TYPE ]                REFRESH   HELP
```

When F4, REFRESH is pressed, the displayed data will be updated.

Table 8.11 FILE memory screen items

Item	Description
Total(KB)	This item displays the amount of total memory available for each device.
Free(KB)	This item displays the amount of free memory available for each device.
Type/Status	This item displays the format type for each device.

9 UTILITY

This chapter explains following special functions 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.

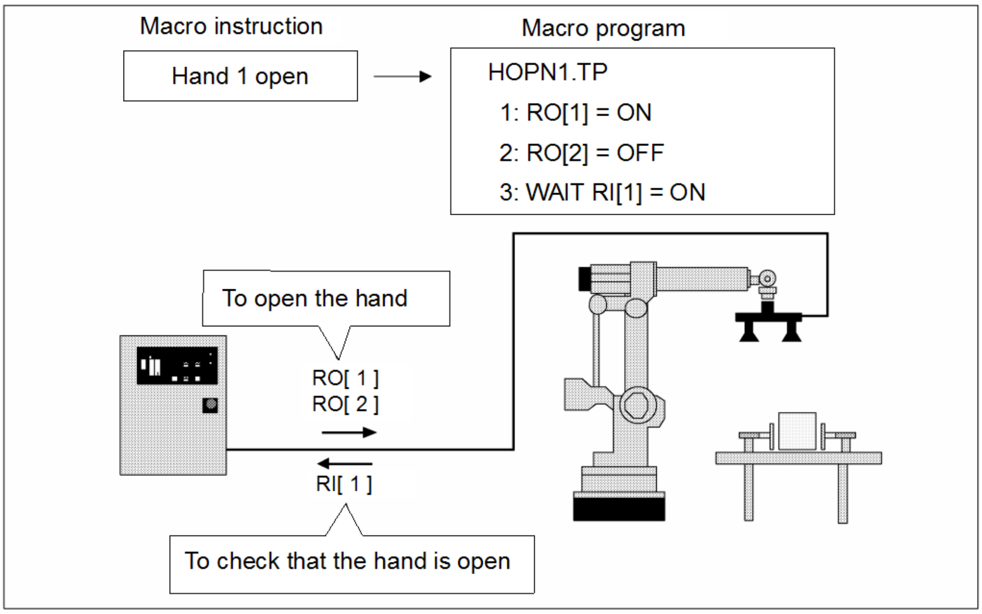


Fig. 9.1 Macro Instructions

A macro instruction has the following capabilities:

- A macro instruction, when taught in a program, can be started as a program instruction.
- A macro instruction can be started using the manual operation screen on the teach pendant.
- A macro instruction can be started using a user key on the teach pendant.
- You can start the macro command using DI, RI, UI, F or M.

Existing programs can be registered as macro instructions. Up to 150 macro instructions can be registered.
A macro instruction can be used according to the following procedure:

1 Create a program to be executed as a macro instruction.
2 Register the created macro program as a macro instruction and determine from which device the macro instruction is to be called.
3 Execute the macro instruction.

The macro instruction setting screen [6 SETUP. Macro] is used for setting a macro instruction.

9.1.1 Setting Macro Instructions

The setting of a macro instruction involves the following items:

- Macro program
- Name of a macro instruction
- Assignment of a device used to start the macro instruction

Macro program

A macro program is a program started by a macro instruction. A macro program can be taught and played back (when played back as a program) in the same way as an ordinary program, except for the following restrictions:

- The subtype of a program, when registered as a macro program, is changed to MR (macro). When the registration of the macro program is canceled, the subtype returns to the original one. (For information about the subtype s, see Subsection 4.1.3.)
- A macro program registered as a macro instruction cannot be deleted.
- A program not including a motion (group) can be started even when the motion enabled state is not set (even when an alarm is issued)（See Motion Group 4.1.4）. For group mask setting, the program information screen is used. (See Subsection 5.3.1.)
- The macro command not having the motion instruction should be made as the program which does not contain the motion group.

Name of a macro instruction

The name of a macro instruction is used to call the macro program from within a program. A macro instruction name must consist of an alphanumeric character string not longer than 36 characters.

> **NOTE**
> Please do not use the parentheses “(” and “)” in the name of a macro instruction.
> Example : HANDOPEN1(HAND1)

Assignment of a device

A macro instruction must be assigned to a key, screen item, etc. so it can be called. The item to which a macro instruction is assigned is called a device. The following devices are available:

- Items on the manual operation screen on the teach pendant (MF)
- User keys on the teach pendant (UK and SU)
- DI, RI, UI, F, M

> **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] : SP cannot be used at present.
- DI[1] to DI[32766] : DI 1 to 32766
- RI[1] to RI[32766] : RI 1 to 32766
- UI[7] HOME signal
- F[1] to F[32766] : F 1 to 32766
- M[1] to M[32766] : M 1 to 32766

> **NOTE**
> 1 The total number of the assign to the DI and RI is up to 10.
> 2 The allocation of macros to UI signals other than the HOME signal can be enabled with system variable $MACRUOPENBL.
> 3 The number which can be actually used is only logical number allocated to the input signal line.

The macro instruction setting screen [6 SETUP. Macro] is used for setting a macro instruction.

> ⚠ **WARNING**
> Before a program set as a macro instruction is copied from a controller onto another controller, the macro setting screens of the two controllers should be compared. It should be ensured that the lists of the two controllers match. The program should be copied only when the lists match. Otherwise, an unpredictable result would be produced.

Procedure 9-1 Setting macro instructions

Condition

- ■ A macro program is created.

```
HOPN1
                                          2/4
      1:   RO[1]=ON
      2:   RO[2]=OFF
      3:   WAIT RI[1]=ON
   [End]

   | POINT |       |       |       | TOUCHUP | >
```

Condition

- Macro program detail information is set.

> **NOTE**
> 1 For greater convenience, a group mask can be set for a program not including motion instructions.
> 2 If the program to be modified contains a motion instruction, the group mask cannot be set.

```
Program detail
                                         1/7
     Creation Date:           16-Jan-1994
     Modification Date:       08-Mar-1994
     Copy Source:

     Positions: FALSE  Size:      176 Byte

     Program name:
      1 HOPN1
      2  Sub Type:       [None             ]
      3  Comment:        [Open HAND1       ]
      4  Group Mask:     [*,*,*,*,*,*,*,*  ]
      5  Write protect:  [OFF              ]
      6  Ignore pause:   [OFF              ]
      7  Stack size:     [     500         ]

        END     DISP     NEXT
```

Changing the motion group (setting a group mask)
Step

1 The program information screen is used to change the group mask.
2 Press [MENU] key to display the screen menu.
3 Select “1 SELECT” on the next page. The program selection screen will be displayed.
4 Press F2, DETAIL on the next page. The program information screen will be displayed.
5 Move the cursor to group 1 of “Group Mask”. Press F5, * to set (*,*,*,*,*,*,*,*).

```
Program detail
                                         4/7

      4  Group Mask:     [*,*,*,*,*,*,*,*  ]

        END     DISP     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 [MENU] 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 will be displayed.

```
Macro Command
                                          1/150
        Instruction name   Program    Assign
     1 [              ][          ]--[     0]
     2 [              ][          ]--[     0]
     3 [              ][          ]--[     0]
     4 [              ][          ]--[     0]
     5 [              ][          ]--[     0]
     6 [              ][          ]--[     0]
     7 [              ][          ]--[     0]
     8 [              ][          ]--[     0]
     9 [              ][          ]--[     0]
    10 [              ][          ]--[     0]

 [ 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.

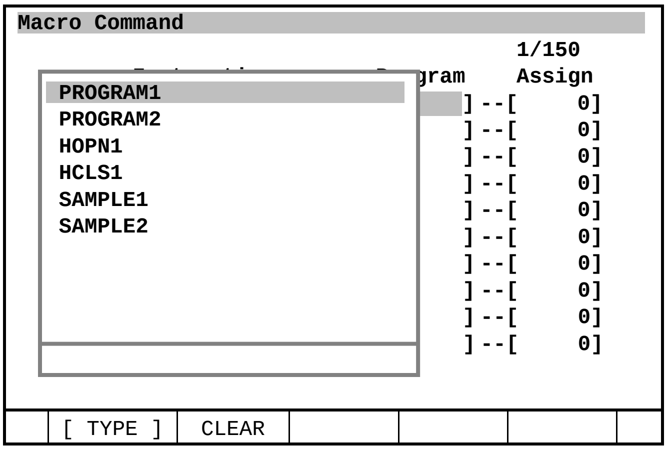
```
Macro Command
                                          1/150
        Instruction name   Program    Assign
     1 [hand          ][          ]--[     0]
     2 [              ][          ]--[     0]
     3 [              ][          ]--[     0]
     4 [              ][          ]--[     0]
     5 [              ][          ]--[     0]
     6 [              ][          ]--[     0]
     7 [              ][          ] Alpha input 1
     8 [              ][          ]   Words
     9 [              ][          ]   Upper Case
    10 [              ][          ]   Lower Case
                                      Options
 abcdef  ghijkl  mnopqr  stuvwx  yz_@*
```

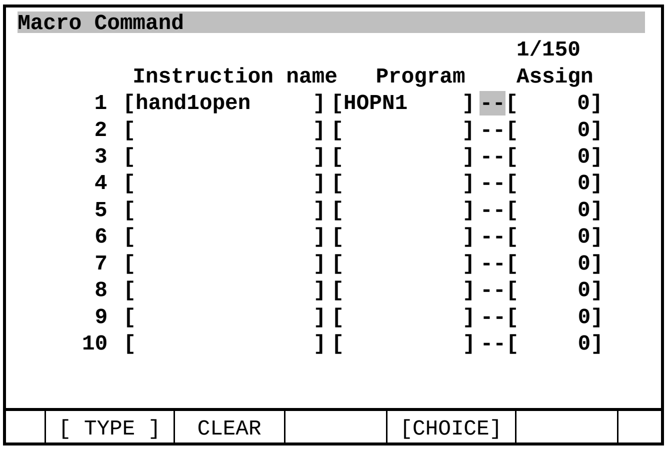
Upon completion of input, press the [ENTER] key.

```
Macro Command
                                          1/150
        Instruction name    Program   Assign
     1 [hand1open     ][          ]--[     0]
     2 [              ][          ]--[     0]
     3 [              ][          ]--[     0]
     4 [              ][          ]--[     0]
     5 [              ][          ]--[     0]
     6 [              ][          ]--[     0]
     7 [              ][          ]--[     0]
     8 [              ][          ]--[     0]
     9 [              ][          ]--[     0]
    10 [              ][          ]--[     0]

 [ TYPE ]  CLEAR           [CHOICE]
```

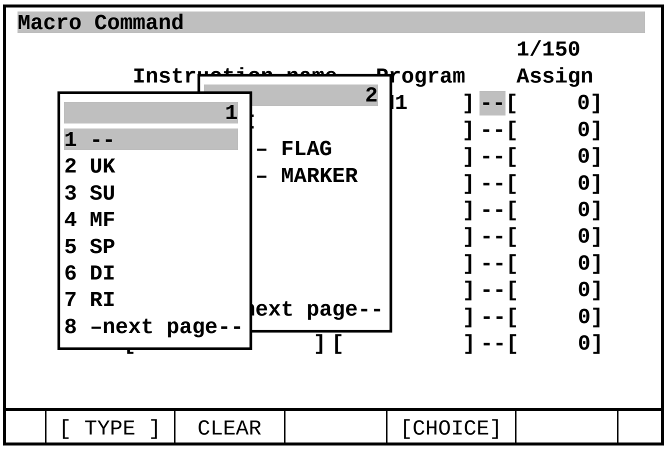
NOTE
No duplicate macro instruction definition is allowed.

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.





7 For device assignment, press F4, [CHOICE] to display a list of device, then choose a device from the list.



```
Macro Command
                                              1/150
       Instruction name   Program     Assign
     1 [hand1open      ] [HOPN1    ] MF[      0]
     2 [               ] [         ] --[      0]
     3 [               ] [         ] --[      0]
     4 [               ] [         ] --[      0]
     5 [               ] [         ] --[      0]
     6 [               ] [         ] --[      0]
     7 [               ] [         ] --[      0]
     8 [               ] [         ] --[      0]
     9 [               ] [         ] --[      0]
    10 [               ] [         ] --[      0]

   [ TYPE ]   CLEAR           [CHOICE]
```

8 Enter a desired device number.

```
Macro Command
                                              1/150
       Instruction name   Program     Assign
     1 [hand1open      ] [HOPN1    ] MF[      1]
     2 [               ] [         ] --[      0]
     3 [               ] [         ] --[      0]
     4 [               ] [         ] --[      0]
     5 [               ] [         ] --[      0]
     6 [               ] [         ] --[      0]
     7 [               ] [         ] --[      0]
     8 [               ] [         ] --[      0]
     9 [               ] [         ] --[      0]
    10 [               ] [         ] --[      0]

   [ TYPE ]   CLEAR           [CHOICE]
```

⚠ **CAUTION**

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

9 For macro instruction deletion, move the cursor to a desired field, then press F2, CLEAR while holding down [SHIFT] key.

```
Macro Command
                                              1/150
       Instruction name   Program     Assign
     1 [hand1open      ] [HOPN1    ] MF[      1]
     2 [hand1close     ] [HCLS1    ] MF[      2]
     3 [               ] [         ] --[      0]
     4 [               ] [         ] --[      0]
     5 [               ] [         ] --[      0]
     6 [               ] [         ] --[      0]
     7 [               ] [         ] --[      0]
     8 [               ] [         ] --[      0]
     9 [               ] [         ] --[      0]
    10 [               ] [         ] --[      0]

   [ TYPE ]   CLEAR
```

10 ”Clear OK?” will be displayed.

- To delete the macro instruction, press F4, YES.

- To cancel deletion of the macro instruction, press F5, NO.

```
Macro Command
                                         1/150
         Instruction name   Program    Assign
      1 [hand1open     ] [HOPN1    ] MF[     1]
      2 [hand1close    ] [HCLS1    ] MF[     2]
      3 [              ] [         ] --[     0]
      4 [              ] [         ] --[     0]
      5 [              ] [         ] --[     0]
      6 [              ] [         ] --[     0]
      7 [              ] [         ] --[     0]
      8 [              ] [         ] --[     0]
      9 [              ] [         ] --[     0]
     10 [              ] [         ] --[     0]
    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 [SHIFT] key held down)
- Pressing user keys on the teach pendant (without pressing [SHIFT] key)
- Pressing user keys on the teach pendant (with [SHIFT] key held down)
- DI, RI, UI, F, M
- Calling the macro instruction from the program

When a macro instruction is started, the macro program is executed in the same way as an ordinary program is executed, except for the following restrictions:

- The single step mode is disabled. The continuous operation mode is always used.
- The macro program is always aborted without the pausing status.
- The macro program is always executed starting from the first line.

When a macro program includes a motion instruction (uses a motion group), the motion enabled state must be set to execute the macro instruction. When no motion group is used, the motion enabled state need not be set.
The motion enabled state is set when:

- ■ ENBL is on.
- ■ SYSRDY output is on. (Servo power supply is on.)

Table 9.1.2 Macro instruction execution conditions

		Without a motion group	With a motion group
MF [1 to 99] SU [1 to 7]	TP enabled	Executable(NOTE)	Executable
UK [1 to 7]		Executable	-
SP [4 to 5] DI [1 to 32766] RI [1 to 32766] UI [7] F [1 to 32766] M [1 to 32766]	TP disenabled	Executable	Executable

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.

*) It is possible to supply an argument in a macro instruction call in a program and use it in a macro program. For details, see Subsection 4.7.6, "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
                                            1/150
         Instruction name   Program    Assign
      1 [hand1open      ] [HOPN1     ] MF[     1]
      2 [hand1close     ] [HCLS1     ] MF[     2]
      3 [               ] [          ] --[     0]
      4 [               ] [          ] --[     0]
      5 [               ] [          ] --[     0]
      6 [               ] [          ] --[     0]
      7 [               ] [          ] --[     0]
      8 [               ] [          ] --[     0]
      9 [               ] [          ] --[     0]
     10 [               ] [          ] --[     0]

   [ TYPE ]   CLEAR
```

Step

1. Press [MENU] 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 will be displayed.

```
MANUAL Macros
                                            1/3
        Instruction
      1 hand1open
      2 hand1close
      3 hand1release



   Press SHIFT-EXEC(F3) to run program

   [ 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 [SHIFT] key. The macro program is started.

Hold down [SHIFT] key until the execution of the macro program is completed.

NOTE
When the macro program contains a motion group, hold down [SHIFT] key until execution of the macro program terminates. If [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 [SHIFT] key is released.

⚠ CAUTION
If [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
                                          1/150
     Instruction name   Program    Assign
  1  [hand1open      ] [HOPN1     ] SU[     1]
  2  [hand1close     ] [HCLS1     ] SU[     2]
  3  [               ] [          ] --[     0]
  4  [               ] [          ] --[     0]
  5  [               ] [          ] --[     0]
  6  [               ] [          ] --[     0]
  7  [               ] [          ] --[     0]
  8  [               ] [          ] --[     0]
  9  [               ] [          ] --[     0]
 10  [               ] [          ] --[     0]

   [ 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 [SHIFT] key.

> **NOTE**
> When the macro program contains a motion group, hold down [SHIFT] key until execution of the macro program terminates. If [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 [SHIFT] key is released during execution, the macro program is terminated forcibly. Note that when execution is interrupted and F3, EXEC is pressed again, the macro program is executed from the first line again.

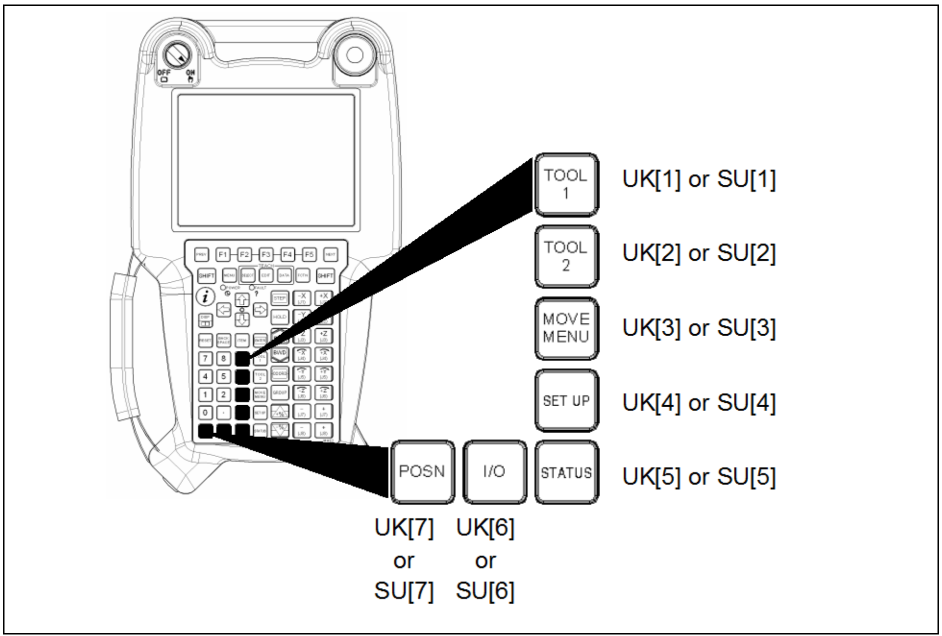


Fig. 9.1.2 User keys on teach pendant

> ⚠ **CAUTION**
> When a key on the teach pendant is assigned to a macro instruction, it becomes that macro instructions device, and the key can no longer be used for its original function.

Procedure 9-4 Execution of macro command using DI,RI, UI, F, M

Condition

■ The teach pendant must be disabled.

■ DI[1 to 32766], RI[1 to 32766], UI[7], F[1 to 32766] or M[1 to 32766] is specified as the device in the macro instruction setting screen.

```
Macro Command
                                        1/150
          Instruction name   Program    Assign
      1 [RET TO REFPOS ][REFPOS    ]UI[     7]
      2 [WORK1 CLAMP   ][CLAMP1    ]DI[     2]
      3 [PROCESS PREP  ][PREP      ]RI[     0]
      4 [              ][          ]--[     0]
      5 [              ][          ]--[     0]
      6 [              ][          ]--[     0]
      7 [              ][          ]--[     0]
      8 [              ][          ]--[     0]
      9 [              ][          ]--[     0]
     10 [              ][          ]--[     0]

   [ TYPE ]   CLEAR
```

Step

1 To start the macro command using DI, RI, UI, F, or M, input the digital signal from the external device or directly input these signals in the I/O screen on the teach pendant.

2 When DI or RI or UI or F or M which is set in the macro instruction setting screen is input, the macro command which is assigned to the signal will be started.

> **NOTE**
> Moreover, $MACROUOPENBL can be changed in the system variable screen displayed at controlled start.

9.2 SHIFT FUNCTIONS

The shift functions shift the specified positions for the operation instructions within a certain range of a previously taught program to other locations.

The shift functions perform the following:

- Shift the position data for the operation instructions within the entire range or within a certain range of an existing program.
- Insert the shift results into a new or existing program.
- Repeat the same shift on another program.

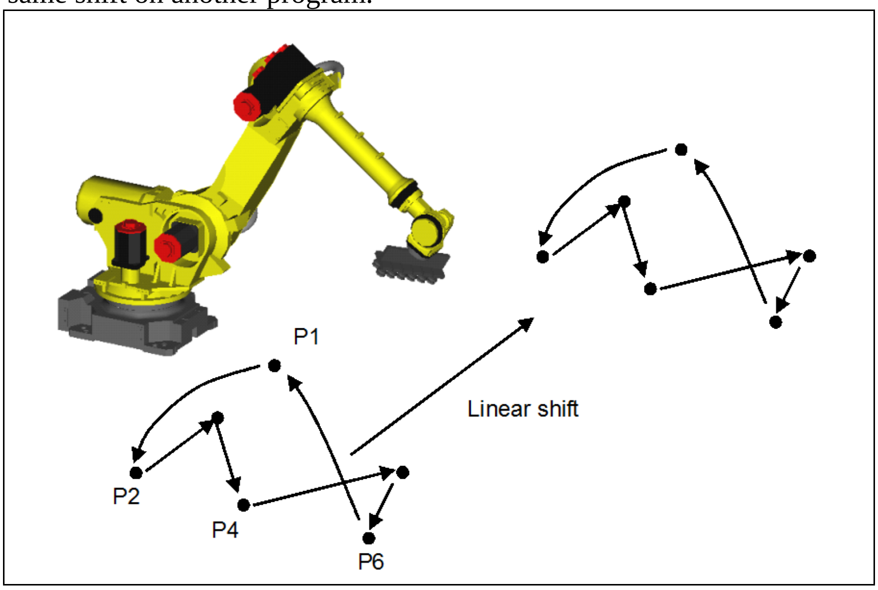


Fig. 9.2 Shift

The following rules apply to converted position data:
Rules governing position data:

- Position data having Cartesian coordinates is converted to Cartesian coordinates. Position data with joint coordinates is converted to joint coordinates.
- If converted joint coordinate position data falls outside the 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 will be displayed 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 entry shift : Performs a rotation shift about a specified rotation axis.

9.2.1 Program Shift Function

The program shift function performs a linear shift or linear rotation shift on the specified positions for the operation instructions within a certain range of a previously taught program.

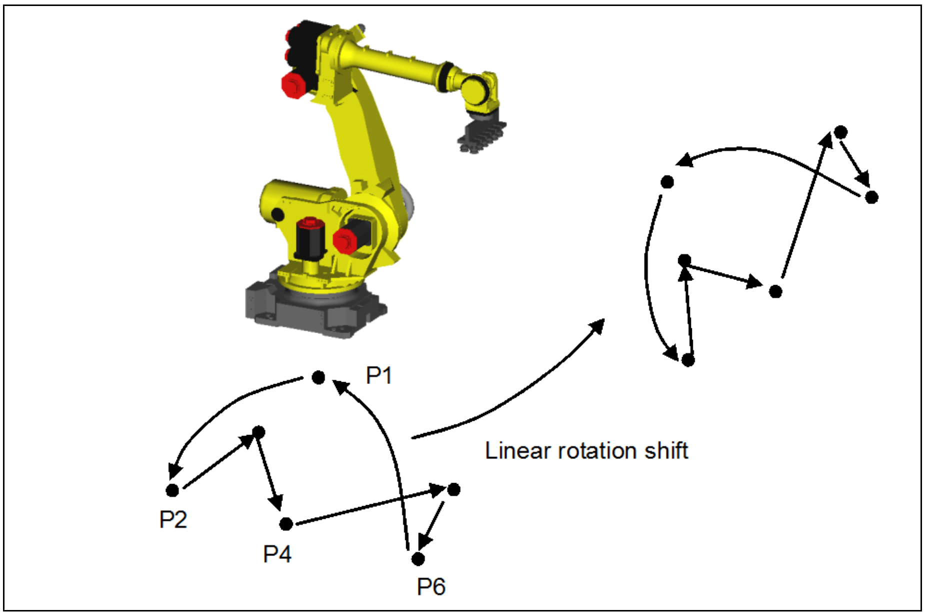


Fig. 9.2.1 (a) Linear rotation shift

The program shift function requires the following setup:

Program name setting

Program name setting specifies the name of the source program, the range of lines on which the shift is to be performed, as well as the name of the program into which the shift results are to be inserted and the line at which they are to be inserted.

Shift information input

Shift information input specifies the direction and amount of the program shift function. Two types of shift are supported: linear shift and linear rotation shift. The shift direction and amount can be specified in either of two ways: representative point specification and direct specification.

- In representative point specification, the user indicates (specifies) representative source and destination points to determine the shift direction and amount.
  For a linear shift, one source point (P1) and one destination point (Q1) must be indicated (specified).

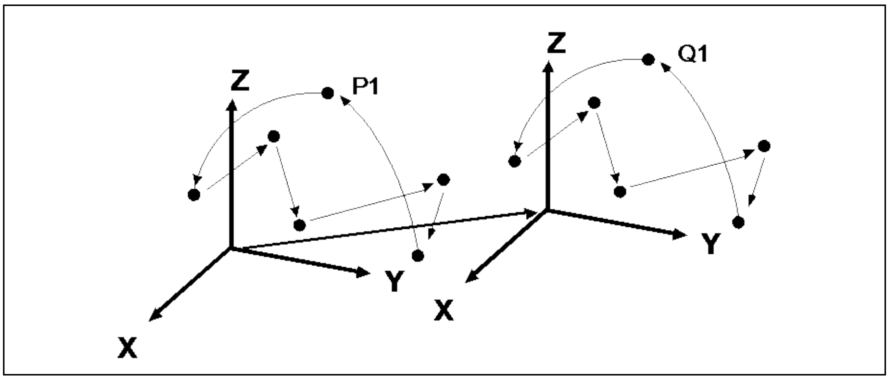


Fig. 9.2.1 (b) Specifying a linear shift

For a linear rotation shift, three source points (P1, P2, and P3) and three destination points (Q1, Q2, and Q3) must be indicated (specified).

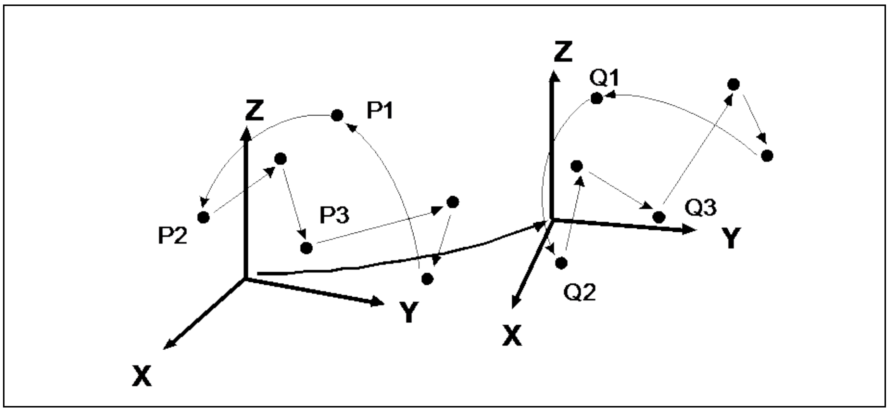


Fig. 9.2.1 (c) Specifying a linear rotation shift

- In direct specification, the user directly specifies the direction and amount (X, Y, Z) of linear shift. In direct specification, linear rotation shift cannot be specified.

To execute the program shift function, use the program shift screen [UTILITIES Program Shift]. The figure below shows how to navigate through the program shift screen.

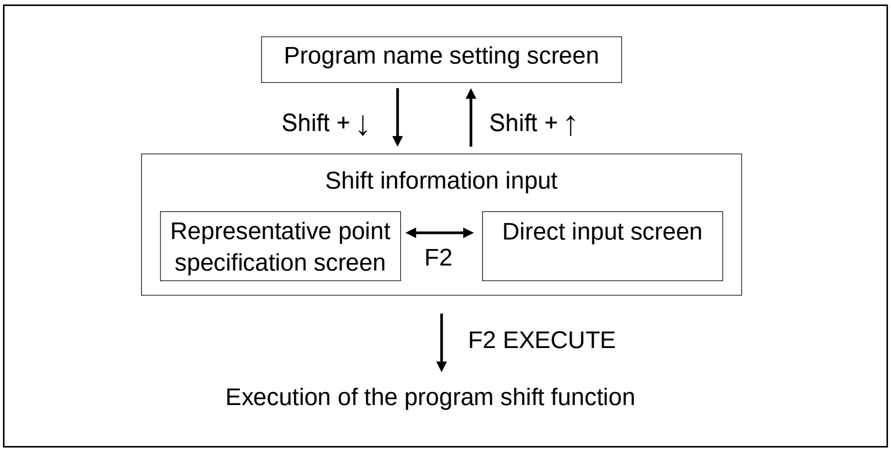


Fig. 9.2.1 (d) Program shift screen

The program name input screen contains the following items:

Table 9.2.1 (a) Contents of the program name input screen

Item	Description
Original Program	Specifies the name of the source program.
RANGE	Specifies the type of the desired range of the source program. ● WHOLE = Performs shift on the entire program. ● PART = Performs shift on part of the program.
Start line	Specifies the start line of the desired range of the source program. If WHOLE is set to all, this item cannot be specified.
End line	Specifies the end line of the desired range of the source program. If WHOLE is set to all, this item cannot be specified.
New Program	Specifies the program into which the shift results are to be inserted. If a new program name is specified, a new program is created with that name. If the name of an existing program is specified, the results are inserted into that program.
Insert line	Specifies the line at which the shift results are to be inserted, if insertion of the results are to be into an existing program is specified. If the program is a new one, this item cannot be specified.
EXT axes	This item is displayed only in the system that has extended axis (E1-3). This item controls the transformation rule of the extended axis. Robot axes only: The position of the extended axis is not changed by the shift transformation. EXT integrated: The position of the extended axis is calculated automatically, the transformation of the extended axis is the same as the robot axes. With EXT axes: (This item is not available for angle entry shift.) The position of extended axis is shifted, the transformation is specified by the extended axis position of P1 and P2. EXT axes only: (This item is not available for mirror image shift and angle entry shift.) The position of extended axis is shifted, and the tool center point of the robot is not changed. The transformation of the extended axis is specified by the extended axis position of P1 and P2. Replace EXT axes: (This item is not available for mirror image shift and angle entry shift.) The position of extended axis is shifted, and the joint position of all robot axes are not changed. The transformation of the extended axis is specified by the extended axis position of P1 and P2.
Motion group	When the original program has two or more motion groups, the motion group page is displayed by pressing [SHIFT] key + down arrow key in the program page. Set the motion group number that will be shifted in this page.

The representative point specification screen contains the following items:

Table 9.2.1 (b) Contents of the representative point specification screen

Item	Description
Position data	Indicates the position of the point where the cursor is currently located. The position is always represented by coordinates in the world coordinate system.
Rotation	Specifies whether rotation is to be performed.
Source position	Specifies the position of a representative source point.
Destination position	Specifies the position of a representative destination point.
REFER	F4 REFER allows the use of a position variable or position register in the source program as the position of a representative point.

Procedure 9-5 Executing the program shift function

Condition

■ The program on which the shift is to be performed exists.

```
TEST2
                                              1/6
        1: J  P[1] 100% FINE
        2: J  P[2] 70% CNT50
        3: L  P[3] 1000cm/min CNT30
        4: L  P[4] 500mm/sec FINE
        5: J  P[1] 100% FINE
     [End]

   POINT                              TOUCHUP    >
```

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select 1, UTILITIES.
3 Press F1, [TYPE]. The screen switching menu will be displayed.
4 Select Program shift. The program name input screen will be displayed.

```
PROGRAM SHIFT
    Program                          1/6
        Original Program :
      1 TEST1
      2   Range:                    WHOLE
      3   Start  line: (not used)   *****
      4   End    line: (not used)   *****
        New Program :
      5 TEST1
      6   Insert line:                  0

    Use shifted up, down arrows for next page

  [ TYPE ]                 [CHOICE]          >
   CLEAR                   [CHOICE]          >
```

5 Specify the necessary items.
6 After specifying the items, go to the next screen with SHIFT + ↓. The representative point specification screen will be displayed. To return to the previous screen, use SHIFT + ↑.

```
PROGRAM SHIFT
    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    >
```

7 For a shift with rotation, set ”Rotation” to ON.

```
PROGRAM SHIFT
    Shift amount/Teach                  1/7
      Position data
      X :********  Y :********  Z :********

      1 Rotation:                        ON
      2 Source position        P1:
      3                        P2:
      4                        P3:
      5 Destination position   Q1:
      6                        Q2:
      7                        Q3:


   [ TYPE ]  EXECUTE            ON      OFF    >
```

8 Specify representative source and destination points.

```
PROGRAM SHIFT
    Shift amount/Teach                5/7
      Position data of Q1
      X :  123.40  Y :  100.00  Z :  120.00

      1 Rotation:                        ON
      2 Source position         P1: Recorded
      3                         P2: Recorded
      4                         P3: Recorded
      5 Destination position    Q1: Recorded
      6                         Q2:
      7                         Q3:


   [ TYPE ]  EXECUTE           REFER   RECORD  >
```

9 For reference point input, press F4, REFER. Select F4 P[] or F5 PR[] to enter arguments.

```
PROGRAM SHIFT
    Shift amount/Teach                      1/7
      Position data of Q1
      X :  123.40  Y :  100.00  Z :  120.00

      1 Rotation:                          ON
      2 Source position         P1: Recorded
      3                         P2: Recorded
      4                         P3: Recorded
      5 Destination position    Q1: Recorded
      6                         Q2:     PR[5]
      7                         Q3:

   [ TYPE ]  EXECUTE         REFER   RECORD  >
```

10 After setting shift information, press F2, EXECUTE and then F4, YES. The conversion results are written into the program.

```
TEST2
                                         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  >
```

11 The direct input screen will be displayed with F2, DIRECT on the next page. Specify the shift amount directly.

```
PROGRAM SHIFT
    Shift amount/Direct entry             1/3

      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.(deg 183)

   deg 183   deg -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.

9.2.2 Mirror Shift Function

The mirror shift function shifts the specified positions for the operation instructions in a certain range of an already taught program symmetrically about a plane.

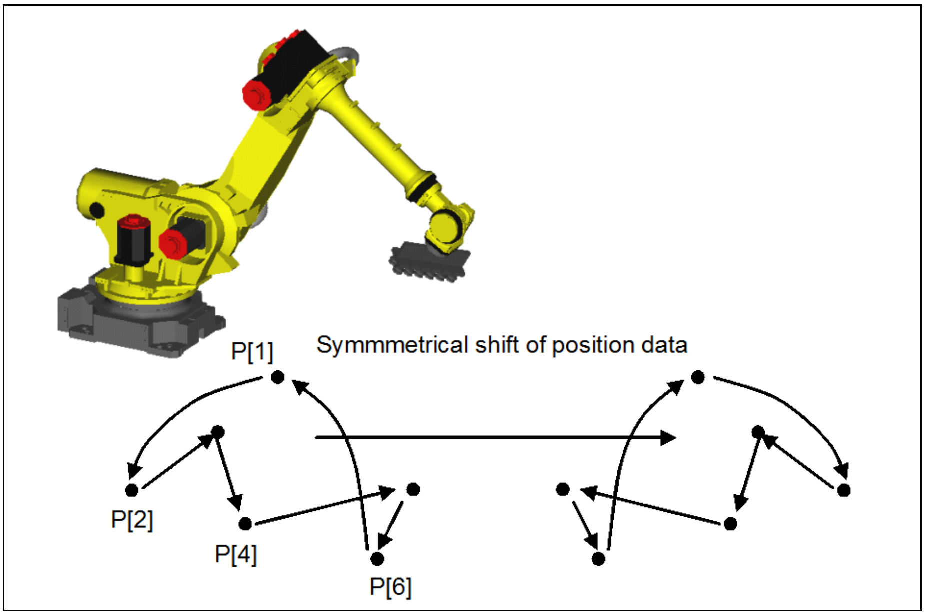


Fig. 9.2.2 (a) Mirror Shift Function

Theoretically, the mirror shift function converts the attitude of the tool from right-handed coordinates to left-handed coordinates. In reality, however, the attitude is returned to the right-handed coordinate system by inverting the Y-axis because no left-handed coordinates exist.
The mirror shift function, therefore, performs conversion most naturally when the plane of symmetry is parallel to the XZ plane of the tool coordinate system.

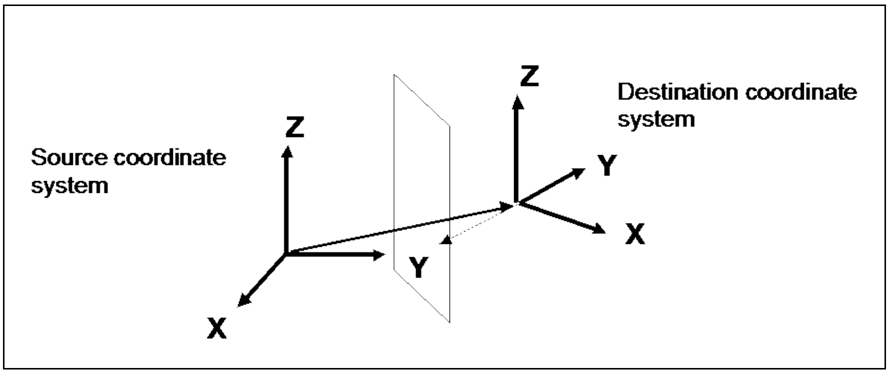


Fig. 9.2.2 (b) Conversion from one tool coordinate system to another with the mirror shift function

> ⚠ **CAUTION**
> 1 The tool coordinate system must be established accurately. The mirror shift function requires that the Z-axis match the tool direction.
> 2 The tool center point (TCP) must be set accurately to ensure correct operation with the points resulting from a symmetrical shift. Otherwise, the points resulting from the shift will contain offset values.

The mirror shift function requires the following setup:

Program name setting

Program name setting specifies the name of the source program, the range of lines on which the shift is to be performed, as well as the name of the program into which the shift results are to be inserted and the line at which they are to be inserted.

Shift information input

Shift information input specifies the direction and amount of the mirror shift. Two types of shift are supported: symmetrical shift and symmetrical rotation shift.

- In representative point specification, the user indicates (specifies) representative source and destination points to determine the shift direction and amount.
  For a symmetrical shift, one source point (P1) and one destination points (Q1), two points in total, must be indicated (specified).
  For a symmetrical rotation shift, three source points (P1, P2, and P3) and three destination points (Q1, Q2, and Q3), six points in total, must be indicated (specified).

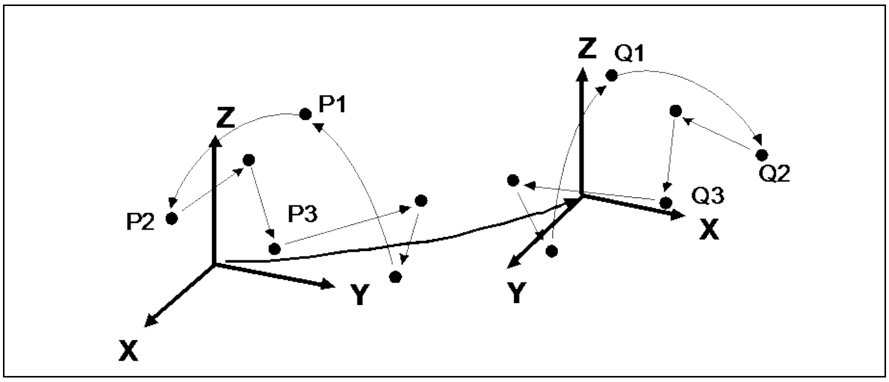


Fig. 9.2.2 (c) Specifying the mirror shift function

To execute the mirror shift function, use the mirror screen [UTILITIES 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
                                               1/6
     1: J  P[1] 100% FINE
     2: J  P[2] 70% CNT50
     3: L  P[3] 1000cm/min CNT30
     4: L  P[4] 500mm/sec FINE
     5: J  P[1] 100% FINE
  [End]

   POINT                          TOUCHUP    >
```

Step

1. Press [MENU] key. The screen menu will be displayed.
2. Select 1, UTILITIES.
3. Press F1, [TYPE]. The screen switching menu will be displayed.
4. Select Mirror Image. The program name input screen will be displayed.

```
MIRROR IMAGE SHIFT
    Program                                    1/6
        Original Program :
      1 TEST1
      2   Range:                             WHOLE
      3   Start  line: (not used)           *****
      4   End    line: (not used)           *****
        New Program :
      5 TEST1
      6   Insert line:                          0

    Use shifted up, down arrows for next page

  [ TYPE ]                  [CHOICE]              >
    CLEAR                   [CHOICE]              >
```

> **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 will be displayed. To return to the previous screen, use SHIFT +↑.

```
MIRROR IMAGE SHIFT
    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     >
```

7 For shift with rotation, set ”Rotation” to ON.

```
MIRROR IMAGE SHIFT
    Shift amount/Teach                1/7
      Position data
      X :********  Y :********  Z :********

      1 Rotation:                      ON
      2 Source position        P1:
      3                        P2:
      4                        P3:
      5 Destination position   Q1:
      6                        Q2:
      7                        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.

> ⚠ **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.

9.2.3 Angle Entry Shift Function

The angle entry 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 entry 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 entry shift function and sets the angular displacement and shift iteration. The representative points can be specified in either of two ways: one in which the rotation axis is specified and one in which it is not specified.

- If the rotation axis is not specified, three representative points (P1, P2, and P3) on the same circumference must be specified. With these three points, the rotation plane and axis are automatically calculated. The intersection of the rotation plane and axis (rotation center) is set as representative point P0.
  Rotation center P0, which is set automatically, can be changed directly later. From the second conversion on, the position of the rotation center can be compensated for by enabling the rotation axis.

- If the rotation axis is specified, a point on the rotation axis must be specified for representative point P0 and any three points on the rotation plane must be specified for representative points P1, P2, and P3. (P1, P2, and P3 need not be on the same circumference.) The rotation plane is determined with representative points P1, P2, and P3. The axis that is vertical to the rotation plane and which passes through representative point P0 is determined as the rotation axis.

In either way, the more distant the representative points P1, P2, and P3, the more precise the conversion. The direction of rotation is regarded as being positive when the rotation is from representative point P1 to P2.

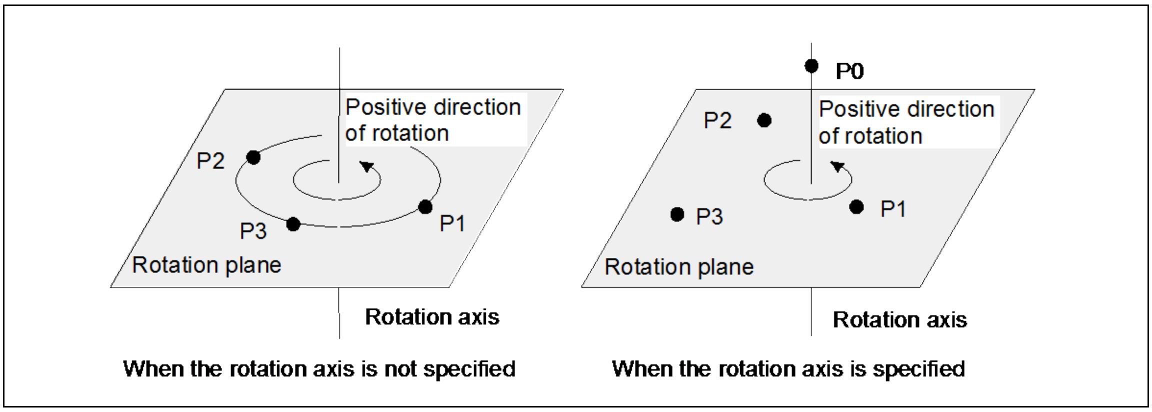


Fig. 9.2.3 (a) Specifying the angle entry shift function

To execute the angle entry shift function, use the angle entry shift screen [UTILITIES Angle Entry Shift]. The figure below shows how to navigate through the angle entry shift screen.

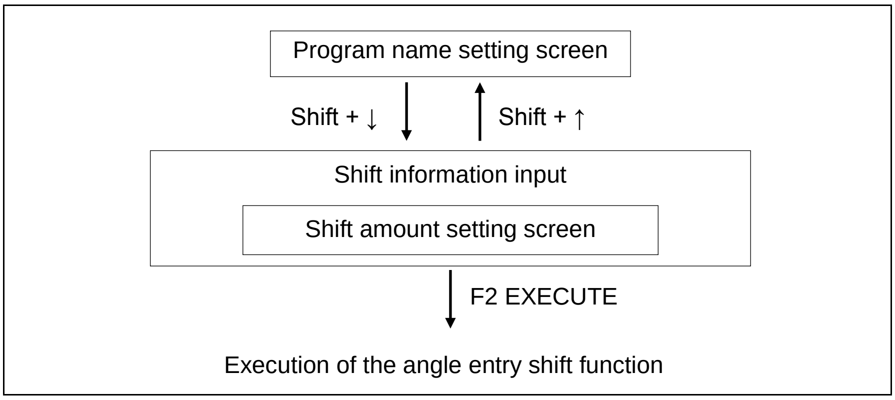


Fig. 9.2.3 (b) Angle entry shift screen

The items on the program name setting screen are the same as those on corresponding screen for the program shift function.
The shift amount setting screen contains the following items:

Table 9.2.3 Contents of the shift amount setting screen

Item	Description
Rotation plane	Specifies the positions of the representative points for determining the rotation plane. If the rotation axis is not specified, these points must be on the same circumference so that the rotation center can be calculated. If the rotation axis is specified, the representative points need not necessarily be on the same circumference. The positions must be specified with coordinates in the world coordinate system.
Rotation axis enable	Specifies how the rotation axis is to be determined from the representative points. The representative points must be specified differently depending on the setting made for this item.
Rotation axis	Specifies the position of representative point P0 for determining the rotation axis. This item is available only when Rotation axis enable is set to TRUE. Only representative point P0 can be specified directly with position data (numeric values) in any coordinate system. To specify P0 directly, position the cursor to this item and press the [ENTER] key. The rotation axis direct specification screen will be displayed.
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 20°, ”iteration” set to 3, and ”destination program” set to program B, program B will be as follows:
	Destination program: Program B
	`1:!Angle entry shift 1 (deg 20.00)`
	`2:J P[1] 100% FINE`
	`3:L P[2] 1500mm/sec FINE`
	`4:!Angle entry shift 2 (deg 40.00)`
	`5:J P[3] 100% FINE`
	`6:L P[4] 1500mm/sec FINE`
	`7:!Angle entry shift 3 (deg 60.00)`
	`8:J P[5] 100% FINE`
	`9:L P[6] 1500mm/sec FINE`
	The position data in program B is as follows:
	P[1]: Position resulting from rotating P[1] in program A by 20°
	P[2]: Position resulting from rotating P[2] in program A by 20°
	P[3]: Position resulting from rotating P[1] in program A by 40°
	P[4]: Position resulting from rotating P[2] in program A by 40°
	P[5]: Position resulting from rotating P[1] in program A by 60°
	P[6]: Position resulting from rotating P[2] in program A by 60°

REFER	F4, REFER allows the use of a position variable or position register in the source program as the position of a representative point.

Procedure 9-7 Executing the angle entry shift function

Condition

- The program on which the shift is to be performed exists.

```
TEST1
                                          1/6
        1: J  P[1] 100% FINE
        2: J  P[2] 70% CNT50
        3: L  P[3] 1000cm/min CNT30
        4: L  P[4] 500mm/sec FINE
        5: J  P[1] 100% FINE
     [End]

   POINT                          TOUCHUP   >
```

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select 1, UTILITIES.
3 Press F1, [TYPE]. The screen switching menu will be displayed.
4 Select Angle entry shift. The program name input screen will be displayed.

```
ANGLE ENTRY SHIFT
    Program                               1/6
        Original Program :
      1 TEST1
      2   Range:                         WHOLE
      3   Start  line: (not used)        *****
      4   End    line: (not used)        *****
        New Program :
      5 TEST1
      6   Insert line:                       0

    Use shifted up, down arrows for next page

  [ TYPE ]                 [CHOICE]          >
    CLEAR                  [CHOICE]          >
```

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 shift amount setting screen will be displayed. To return to the previous screen, use SHIFT + ↑.

```
ANGLE ENTRY SHIFT
    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                                    >
```

7 For shift with the rotation axis specified, set ”Rotation axis specification” to TRUE. If required, specify ”Iteration”.
8 Specify the representative points.

```
ANGLE ENTRY SHIFT
    Shift amount                      1/7
      Position data of P1
      X :  123.40  Y :  100.00  Z :  120.00

      1 Rotation plane         P1: Recorded
      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                                    >
```

9 For reference point input, press F4, REFER. Select F4, P[ ] or F5, PR[ ] to enter arguments.

```
ANGLE ENTRY SHIFT
    Shift amount                      2/7
      Position data of P2
      X :  123.40  Y :  135.00  Z :   98.20

      1 Rotation plane         P1: Recorded
      2                        P2:      P[5]
      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                                    >
```

10 Enter the angular displacement.
11 After setting the shift information, press F2, EXECUTE to execute the shift.
12 If the turn number is changed due to the conversion, the user is notified and prompted to make a selection.

```
    Repeat3:Select P[3]:J6.(deg 183)

  deg 183 | deg -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 will be displayed.

```
ANGLE ENTRY SHIFT
    Shift amount                      1/4
    Rotation center axis direct entry
      1 Frame                   User Frame 1
      2 X (mm):                         0.00
      3 Y (mm):                         0.00
      4 Z (mm):                         0.00

  [ TYPE ] | EXECUTE |        | [CHOICE] |        | >
  CLEAR                         [CHOICE]            >
```

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.

16 Provide the other necessary shift information has been set, press F2, EXECUTE to execute the shift.

17 To erase all the shift information, press F1, CLEAR on the next page.

9.3 COORDINATE SYSTEM CHANGE SHIFT FUNCTIONS

The coordinate system change shift functions change the tool coordinate system (tool) or user coordinate system for the operation instructions within a certain range of an already taught program, and convert the position data so that the TCP is located at the same position, considering the shift amount resulting from the change from the old to the new coordinate system.

> **NOTE**
> The coordinate system change shift functions allow the user to specify that the position data not be converted.

Coordinate system change shift functions

The coordinate system change shift functions perform the following:

- Change the tool coordinate system or user coordinate system number in the position data (Cartesian coordinates) for the operation instructions within the entire range or within a certain range of an existing program.
- If the position data is joint coordinates, convert the coordinates considering the shift amount resulting from the tool change or user coordinate system change.
- Insert the shift results into a new or existing program.
- Execute the same shift on another program.

Position data conversion

The following rules apply to converted position data:
Rules for positions and attitudes:

- Position data with Cartesian coordinates is converted to Cartesian coordinates. Position data with joint coordinates is converted to joint coordinates.
- If converted joint coordinate position data falls outside the 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 center 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 [UTILITUES Tool Offset (Frame Offset)]. The figure below shows how to navigate through the change shift screen.

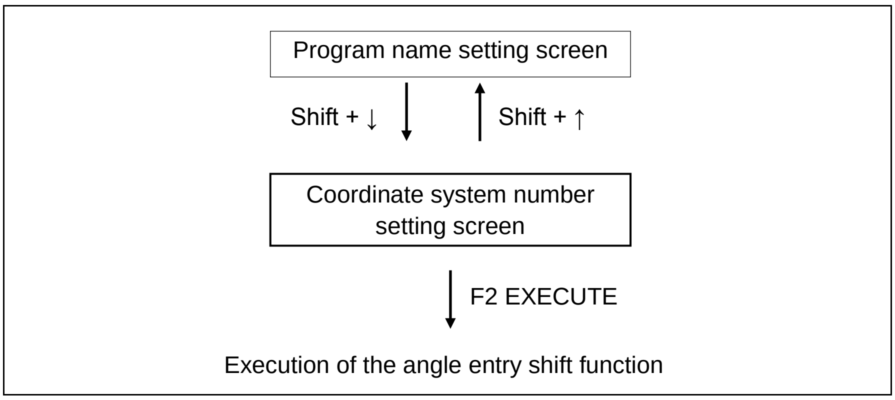


Fig. 9.3 Coordinate system shift screen

Procedure 9-8 Executing the tool change shift function

Condition

- The program on which the shift is to be performed exists.

```
TEST1
                                            1/6
     1: J  P[1] 100% FINE
     2: J  P[2] 70% CNT50
     3: L  P[3] 1000cm/min CNT30
     4: L  P[4] 500mm/sec FINE
     5: J  P[1] 100% FINE
  [End]

 | POINT |  |  |  | TOUCHUP | >
```

Step

1. Press [MENU]. The screen menu will be displayed.
2. Select 1 UTILITIES.
3. Press F1, [TYPE]. The screen switching menu will be displayed.
4. Select Tool Offset. The program name input screen will be displayed.

```
TOOL OFFSET
    Program                              1/6
        Original Program :
      1 TEST1
      2   Range:                        WHOLE
      3   Start  line: (not used)       *****
      4   End    line: (not used)       *****
        New Program :
      5 TEST1
      6   Insert line:                      0

    Use shifted up, down arrows for next page

 | [ TYPE ] |  |  | [CHOICE] |  | >
     CLEAR                 [CHOICE]          >
```

5. Specify the necessary items.
6. After specifying the items, go to the next screen with SHIFT + ↓. The coordinate system number setting screen will be displayed. To return to the previous screen, use SHIFT + ↑.

```
TOOL OFFSET
    UTOOL number                           1/3

      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.
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.(deg 183)

| deg 183 | deg -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 NEXT ">" and press F1, CLEAR on the next page.

> ⚠ **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
                                        1/6
      1: J  P[1] 100% FINE
      2: J  P[2] 70% CNT50
      3: L  P[3] 1000cm/min CNT30
      4: L  P[4] 500mm/sec FINE
      5: J  P[1] 100% FINE
   [End]

| POINT |  |  |  | TOUCHUP | > |
```

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select 1, UTILITIES.
3 Press F1, [TYPE]. The screen switching menu will be displayed.
4 Select Frame Offset. The program name input screen will be displayed.

```
UFRAME OFFSET
  Program                              1/6
     Original Program :
   1 TEST1
   2   Range:                         WHOLE
   3   Start  line: (not used)        *****
   4   End    line: (not used)        *****
     New Program :
   5 TEST1
   6   Insert line:                       0

  Use shifted up, down arrows for next page

| [ TYPE ] |  |  | [CHOICE] |  | > |
  CLEAR                [CHOICE]           >
```

5 Specify the necessary items.

6 After specifying the items, go to the next screen with SHIFT + ↓. The coordinate system number setting screen will be displayed. To return to the previous screen, use SHIFT + ↑.

```
UFRAME OFFSET
    UFRAME number                        1/3

      1 Old UFRAME number:                 0
      2 New UFRAME number:                 0
      3 Convert Position data (Y/N):      NO

   [ 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.
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.(deg 183)

  deg 183 | deg -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 NEXT ”>” and then press F1, CLEAR on the next page.

> ⚠ **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 POSITION REGISTER LOOK-AHEAD EXECUTION FUNCTION

While the robot is executing a program, it reads the lines ahead of the line currently being executed (look-ahead execution).
Conventionally, look-ahead execution was performed for motion statements having normal position data (not using position registers). Look-ahead execution could not be performed for motion statements that used position registers for their position data.
Motion statements using position registers could not be read in advance because the values in the position registers could be changed by the program, data transfer function, and so forth.
* If the robot reads a motion statement using a position register prior to its execution, the value of the position register may yet be changed by a program or another function (such as data transfer). Such a change is not reflected in the motion statement that has already been read by the robot. Consequently, the robot’s operation may be unpredictable.
Motion statements that use position registers can be classified into two types:
- Motion statements with the target position specified by a position register
- Motion statements with an offset instruction where an offset is given by a position register

Even when a target position or offset is calculated during program execution, and a position register holding this calculation result is used with a motion statement, look-ahead execution was not performed for the statement, for the reason explained above.
The position register look-ahead execution function enables look-ahead execution for position registers. For this purpose, an instruction to lock position registers and an instruction to unlock the registers are newly provided. By means of these instructions, the user can explicitly specify a program portion. Then, for the specified program portion, even when it contains motion statements that use position registers, look-ahead execution can be performed.

Function

The position registers can be locked to prevent their contents from being changed after they are read. When an attempt is made to execute an instruction to change a locked position register (for example, an assign instruction for the position register, or an application instruction to set data in the position register), the following alarm message is issued:
“INTP-128 Pos reg is locked”

When a function (such as the data transfer function) other than the program attempts to change the value of a locked position register, the following alarm message is issued, and the attempt fails:
“VARS-037 Pos reg is locked”

Position registers are generally locked and unlocked with instructions taught in a program. When a program that has locked the position registers terminates, the position registers are unlocked automatically.
* In case current program is called by other one, program execution will be returned to program that calls current program when END instruction is executed. In this case, the position registers are not unlocked because program execution is not terminated yet.

All position registers are locked simultaneously. While the position registers are locked, access to any position register is disabled, even in a different motion group.

> **NOTE**
> Before using position register instructions, lock position registers. When position register instructions are used with the position registers unlocked, operation may become tight.

Operation

The following program instructions have been added:

- LOCK PREG

Locks all position registers. This instruction prevents any change being made to any position register.

- UNLOCK PREG

Unlocks the position registers.
These are control instructions (not motion instructions). They can be taught in the same way as other control instructions (See Subsection 5.3.5, ”Teaching a Control Instruction”).

Example

The following shows how to use the LOCK PREG and UNLOCK PREG instructions in a program:

```
1: J P[1] 100% FINE
2:   PR[1]=PR[2]
3:   PR[2]=PR[3]
4: LOCK PREG
5: L P[2] 100mm/sec Cnt100
6: L P[3] 100mm/sec Cnt100
7: L PR[1] 100mm/sec Cnt100
8: L P[4] 100mm/sec Cnt100 offset, PR[2]
```

```
 9: L P[5] 100mm/sec FINE
10: UNLOCK PREG
```

When line 4 of this sample program has been executed, the position registers are locked. They are unlocked when line 10 has been executed. Therefore, the motion statements with position registers in lines 7 and 8, which are executed with the position registers locked, are subject to look-ahead execution.
If the program is terminated between lines 4 and 10, the locked position registers are unlocked automatically.
If the program is paused between lines 4 and 10, the cursor is moved manually, then the program is restarted, the locked position registers are unlocked. In this case, look-ahead execution is not performed for the statements in lines 7 and 8.

> **NOTE**
> When 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.)

> **NOTE**
> 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.

9.5 TIME BEFORE FUNCTION

This function calls program or outputs signal at specified timing before or after the completion of motion instruction.

This function can reduce waiting time for communication to/from external device and improve cycle time.

Function

This function can call subprogram or output signal at specified timing before or after the completion of motion instruction.

Using an instruction in a program, specify the time at which instruction part is to be executed (in seconds).

The timing of motion instruction completion is defined to be 0 second, which differs depending on the termination type (FINE or CNT).
The TIME BEFORE (or AFTER) instruction is a motion option instruction. Both execution timing and instruction part should be specified with the motion option instruction.

Instruction statement

Specify the execution timing and the instruction part after the motion instruction.
Following instruction can available for instruction part.

- CALL program
- Signal output
- Point Logic (Refer to ‘9.7 Point Logic Instruction’ for details.)

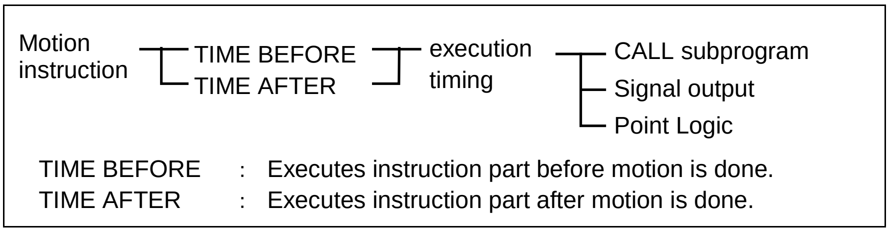


Fig. 9.5 (a) TIME BEFORE/AFTER instruction (motion option instruction)

Example

```
1: J  P[1] 100% FINE
 :   TB 1.00sec CALL OPENHAND

1: J  P[1] 100% FINE
 :   TA 1.00sec,DO[1]=ON

1: J  P[1] 100% FINE
 :   TA 1.00sec POINT_LOGIC
```

Execution timing

According to the specified execution timing, the instruction part is executed at the following timing:
If execution timing, ”n” seconds, is specified with a TIME BEFORE instruction, the instruction part is executed n seconds before completion of motion instruction.

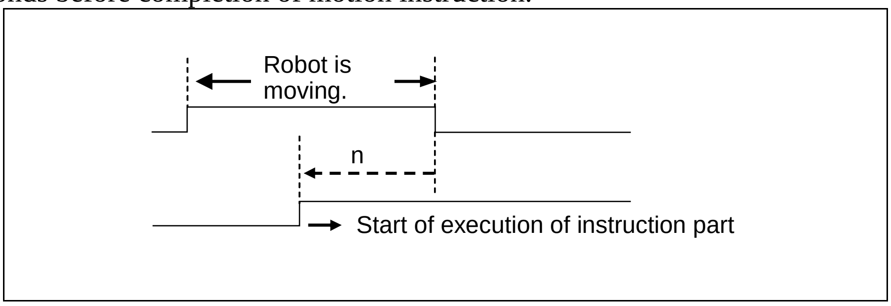


Fig. 9.5 (b) Timing sequence (time before instruction)

If execution timing, ”n” seconds, is specified with a TIME AFTER instruction, the instruction part is executed n seconds after completion of motion instruction.

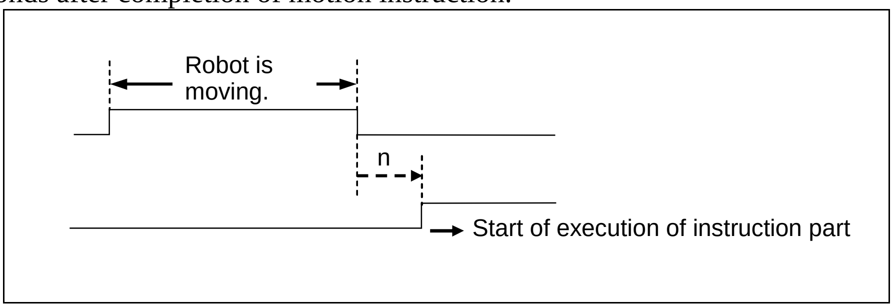


Fig. 9.5 (c) Timing sequence (time after instruction)

If the execution timing specified with a TIME BEFORE instruction is earlier than motion start timing, the instruction part is executed as soon as motion starts.

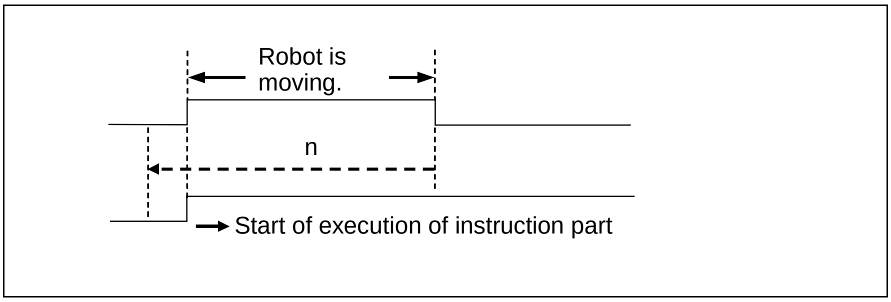


Fig. 9.5 (d) Timing sequence (time before instruction)

The execution timing that can be specified in a program is
- 0 to 30 seconds for a TIME BEFORE instruction
- 0 to 0.5 seconds for a TIME AFTER instruction

> ⚠ **CAUTION**
> When override is changed, moving distance of the robot during specified time is also changed. So the position that interaction part is executed can be changed depending on override.

Finding/Replacing Instructions

- Finding Instruction

You can find program which is used for TIME BEFORE/AFTER by “find” on F5 pull-up menu. Select "CALL" and then "Call program" to find program used in TIME BEFORE/AFTER.
You can find signal output instruction for TIME BEFORE/AFTER too. Select item “I/O” on submenu.

- Replacing Instruction

- TIME BEFORE/AFTER can be replaced to TIME BEFORE/AFTER and DISTANCE BEFORE by selecting “replace” on F5 pull-up menu. Select “TIME BEFORE/AFTER” on replace item submenu.
- You can replace subprogram of TIME BEFORE/AFTER by selecting “CALL” on replace item submenu.
- You can replace signal output of TIME BEFORE/AFTER by selecting “I/O” on replace item submenu.

Single step

When motion instruction with TIME BEFORE/AFTER CALL subprogram is executed in single-step mode, motion stops temporarily at the time when the subprogram is called. Subsequently, the rest of the motion is executed in sync with single-step execution of the subprogram.

Power failure handling

If power failure handling is enabled and the power is turned down 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 at the position the robot was located when the power was turned down. Execution timing of subprogram is different from the usual timing. You should take care of this point.

Procedure 9-10 Entering the TIME BEFORE instruction

Step

1 Move cursor to motion option area.

```
PNS0001
                                1        1/2
        1: J  P[1] 100% FINE
     [End]



                          [CHOICE]
```

2 Press function key F4, [CHOICE]. A list of motion option instructions will be displayed.

```
Motion Modify   1      Motion Modify   2      Motion Modify   3
1 No option            1 Incremental          1 TIME AFTER
2 Wrist Joint          2 Tool_Offset          2 DISTANCE BEFORE
3 ACC                  3 Tool_Offset,PR[      3 PTH
4 Skip,LBL[]           4 Independent EV       4
5 BREAK                5 Simultaneous EV      5
6 Offset/Frames        6 TIME BEFORE          6
7 Offset,PR[ ]         7 Skip,LBL,PR          7
8 --next page--        8 --next page--        8 --next page--
```

3 Select item TIME BEFORE.

```
PNS0001
                                         1/2
        1: J  P[1] 100% FINE TB   0.00sec ...
         :
     [End]



    Enter value
   REGISTER               [CHOICE]
```

4 Specify the time and press the [ENTER] key. Example: 2 seconds.

```
PNS0001
                                         1/2
   TIME statement 1 100% FINE TB   2.00sec ...
   1 CALL program
   2 CALL program()
   3 DO[ ]=...
   4 RO[ ]=...
   5 GO[ ]=...
   6 POINT_LOGIC
   7 AO[ ]=...
   8
        1: J  P[1] 100% FINE TB   2.00sec ...
                          [CHOICE]
```

5 Select item “CALL program”.
Select item “CALL program ( )” to use AR.

```
PNS0001
                                          1/2
 OPEN_HAND                           2.00sec,
 CLOSE_HAND



 PROGRAM  MACRO  KAREL  INDIRECT  STRINGS  >
```

6 Select item OPEN_HAND.

```
PNS0001
                                          1/2
        1: J  P[1] 100% FINE TB   2.00sec,
         :  CALL OPEN_HAND
     [End]



                          [CHOICE]
```

Program example

Main program: PNS0001

```
1:J  P[1] 100% FINE
2:J  P[2] 100% CNT100 TB  1.00sec,
 :  CALL OPEN_HAND
3:  CALL CLOSE_HAND
```

Subprogram: OPEN_HAND

```
1:  DO[1]=ON
```

Behavior when the main program is executed

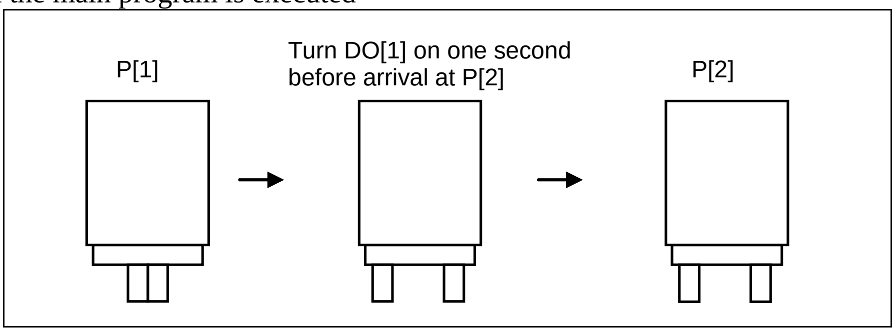


Fig. 9.5 (e) Program example using a TIME BEFORE instruction

Notes/restrictions

In the subprogram specified for Call, motion instructions cannot be specified. (The motion group in the subprogram must be [*, *, *, *, *, *, *, *].)

Since the called sub routine and the main program are concurrently executed, the main program is sometimes executed earlier than the called sub routine.

When you do not want to proceed with the execution of the main program ahead until the execution of the called sub routine is finished, please change a system variable as follows.

$TIMEBF_VER=3 (the standard value) → 2

There is no limit to the number of lines in the subprogram.

The TIME BEFORE/AFTER motion option instructions cannot be used with DISTANCE BEFORE.

The TIME BEFORE/AFTER motion option instructions cannot be used with application instructions such as spot[ ] and skip instructions.
If the termination type of the motion instruction is CNT, the timing of motion completion changes depending on the degree of CNT. The timing at which the subprogram is called changes accordingly.
Depending on the situation, even if the execution timing is set to 0 seconds with a TIME BEFORE instruction, the subprogram may be executed too early. In this case, use a TIME AFTER instruction.

For direct specification of signal output, only DO, RO, GO, and AO are supported.

If the override is modified dynamically while the motion instruction which has TIME BEFORE instruction is executed, the timing at which the subprogram is called may not be proper.

When there is TIME BEFORE/AFTER (TB) or DISTANCE BEFORE (DB) that is not yet triggered in previous line, TB/DB does not trigger even when trigger condition is satisfied. (TB/DB always triggers after previous TB/DB triggered.)

9.6 DISTANCE BEFORE FUNCTION

9.6.1 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 <u>DB 100mm,CALL A</u>

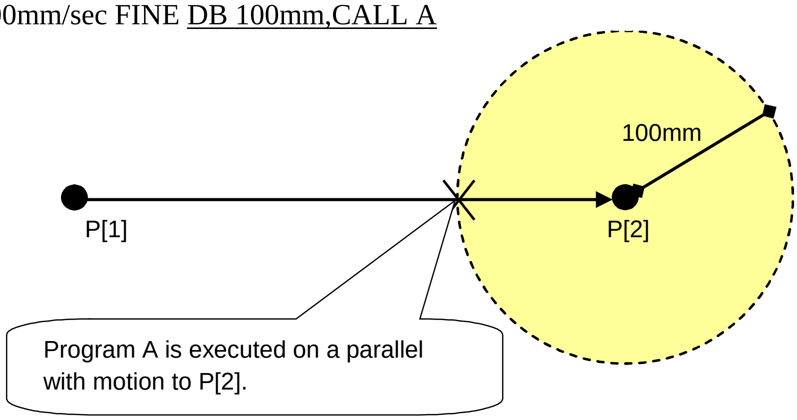


Fig. 9.6.1 Execution timing of distance before

9.6.2 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 (NOTE)	TCP goes into a region, which is within specified distance from destination point. Please refer to 9.6.4 Instruction 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 ● Point Logic	Program to be called cannot use motion group. Only logic instruction is available.

> **NOTE**
> This is condition to process instruction part.

9.6.3 Configuration

Before using Distance Before, set following system variable.

$SCR_GRP[1].$M_POS_ENB = TRUE

9.6.4 Instruction

1 Format

Distance Before is taught in following format.

```
Motion statement + DB distance value, instruction part
```

Example

```
L P[2] 1000mm/sec FINE DB 100mm, CALL A
```

Instruction part (Please refer to 3.)

Distance value (Please refer to 2.)

> **NOTE**
> Distance Before is a motion option. You cannot use DB as a standard instruction.

2 Distance value

(i) Distance value

Distance Before executes instruction part when TCP goes into a spherical region whose center is destination point. Distance value decides the radius of this sphere. Distance value is taught in millimeter. Distance value is from 0 to 999.9mm. This sphere is referred as trigger region hereafter.

```
1:   L P[1] 2000mm/sec FINE DB 100.0mm  DO[1] = ON
2:   L P[2] 2000mm/sec FINE DB 100.0mm  DO[1] = ON
```

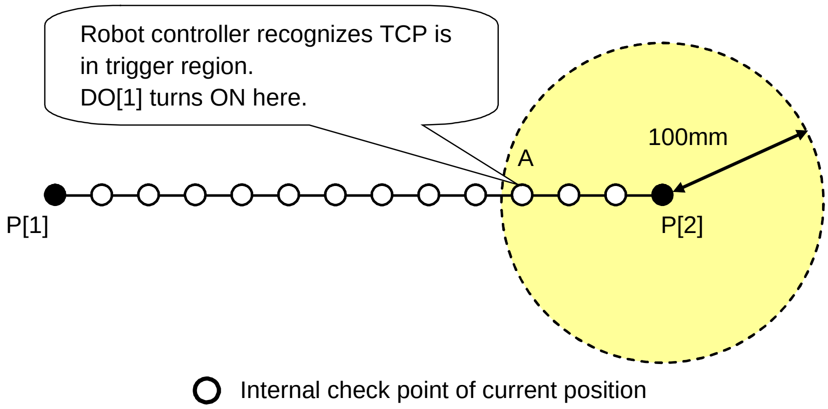


Fig. 9.6.4 (a) Cyclical checks if TCP goes into trigger region.

Internally, Robot controller calculates current position to judge if TCP is in trigger region or not. Instruction part is executed when this calculated position is in trigger region.

> ⚠ **CAUTION**
> Execution timing of instruction part is decided by distance (in millimeter). Because judgment to trigger is done by calculating distance between current position and destination point, actual execution timing is different from distance value.
> (Error in case of 2000mm/sec is estimated around 16mm.)

(ii) Radius of trigger region.

Radius of trigger region is as follows.
Radius = (distance value or $DB_MINDIST) + $DB_TOLERENCE

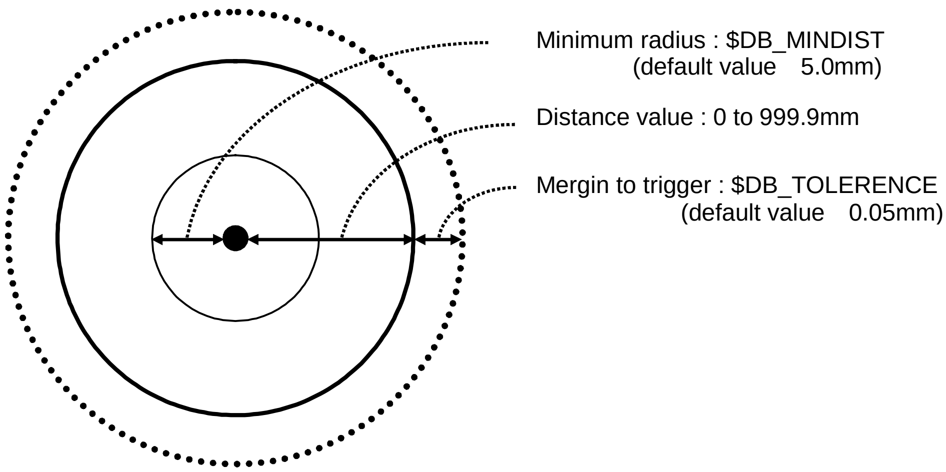


Fig. 9.6.4 (b) The size of trigger region

If distance value is less than $DB_MINDIST, $DB_MINDIST is used as distance value.
Example Suppose following motion statement is taught with $DB_MINDIST = 5.0

```
L P[1] 2000mm/sec FINE DB 0.0mm DO[1]=ON
```

In this case, Robot controller interprets it as DB 5.0mm. Then $DB_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
- Point Logic

(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[ ]

Example)

```
L P[2] 1000mm/sec FINE DB 100mm, DO[1]=ON
```

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.

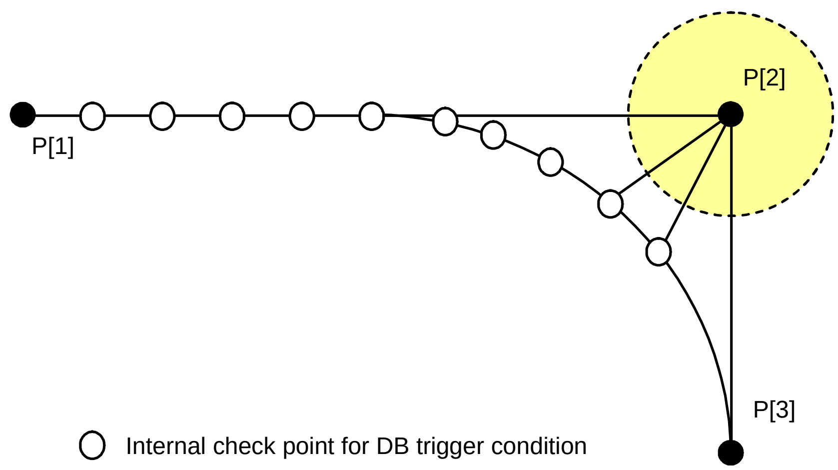
(iii) Point Logic

Refer to '9.7 Point Logic Instruction' for details.

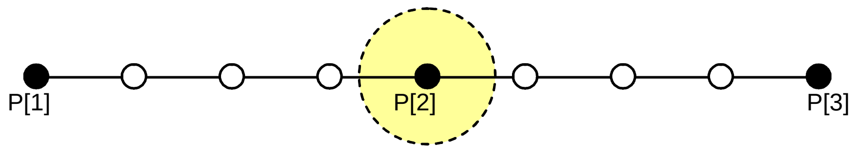
4 Changing trigger condition

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 (NOTE 2)	“going away” +“penetrate” +end of motion (NOTE 2)
1 (default value)	“region trigger” +“going away” +“penetration” +end of motion (NOTE 2)	end of motion (NOTE 2) +(“going away”) (NOTE 1)
2	“region trigger” +“penetration” +end of motion (NOTE 2)	“going away” end of motion (NOTE 2)

“going away” and “penetration ” is defined in (i), (ii) and (iii) respectively.
Distance Before executes instruction part when DB condition is satisfied. Otherwise, posts alarm.
There are two alarms for not-triggered DB. They are INTP-293 and INTP-295. $DBCONDTRIG decides which alarm is posted. Message is same but severity is different. Please refer to 5 for details.

NOTE

1 When Distance Before is triggered by “going away” in case of $DB_CONDTYP = 1, you can post alarm in addition to execution of instruction part. Please refer to 4 (i) for details.

2 By default configuration, if motion statement with Distance Before completes and robot stops before neither “region trigger” nor “going away” nor “penetration” trigger happens, Distance Before executes instruction part and post alarm. Please refer to 4 (iii).

(i) In case of going away.

If termination type is CNT and distance value is small, TCP may not go into trigger region.

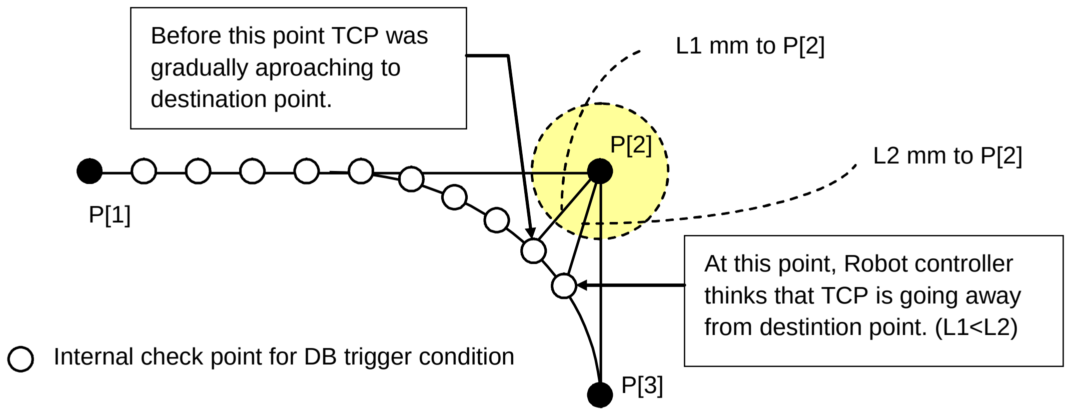


Fig. 9.6.4 (c) TCP doesn’t go into trigger region.

In case of Fig. 9.6.4 (c), TCP doesn't go into trigger region. TCP starts to go away from destination point (P[2]). Robot controller cyclically judges if TCP is going away from destination point or not in addition to DB condition. Robot controller recognizes that TCP is going away when calculated distance between current position and destination point is greater than distance between destination point and nearest point by more than ($DB_AWAY_TRIG) millimeter. This case is referred as "going away" in this manual.

- To post alarm in addition to execution of instruction part only when the DB is triggered by "going away" trigger, set $DB_AWAY_ALM to TRUE. DB executes instruction part and post following alarm.
  INTP-295 (program name, line number) DB too small (away) (mm)
  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.9.6.4 (d).

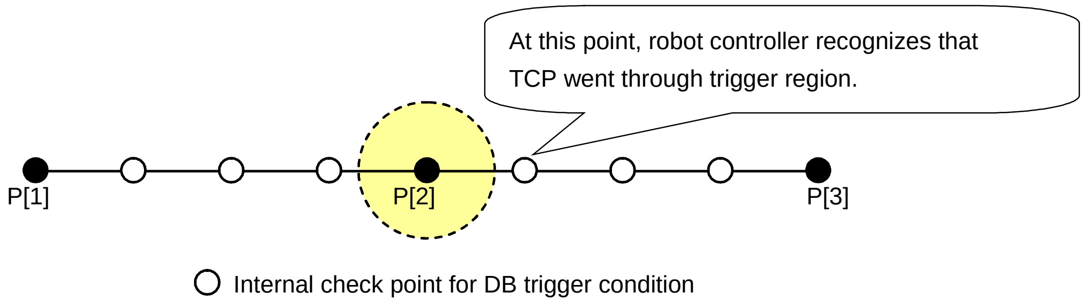


Fig. 9.6.4 (d) Penetration

In this case, TCP moves too fast for robot to check DB condition in small trigger region. Because cyclical check is done outside of trigger region, the fact TCP is in trigger region is not recognized by the robot controller. This case is referred as " penetration" in this manual.
To handle cases like Fig. 9.6.4 (d), Distance Before checks if TCP went through trigger region or not. If trajectory of TCP penetrated trigger region (penetration), instruction part is executed by default configuration.
But in this case, execution of instruction part is done after TCP passed away destination point.

- Motion with termination type FINE doesn't cause trigger by "penetration".

(iii) End of motion

If motion statement with DB completes and robot stops before "region", "going away" and "penetration" is satisfied, DB executes instruction part and post following alarm.
INTP-297 (program name, line number) DB too small (done) (mm).
This alarm is not posted by FINE motion.
If you don't want this trigger, set $DB_MOTNEND to FALSE (default value: TRUE).
Distance displayed by this alarm is distance to destination.

> ⚠ **CAUTION**
> 1 If you stop your robot by E-stop when motion statement is about to complete, Distance Before may be trigger just after resume of the program.
> 2 If you halt a program when motion statement with DB is near its completion, DB may not be triggered. In this case, Distance Before executes its instruction part after resume of program.

5 Alarms for not -triggered Distance Before

Distance Before posts alarm if condition is not triggered. What is posted depends on $DBCONDTRIG.

$DBCONDTRIG	Alarm to be posted
0 (default value)	INTP-295 WARN (Program name, line number)DB condition was not triggered. (Distance mm)
1	INTP-293 PAUSE.L (Program name, line number)DB condition was not triggered. (Distance mm)

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). For the reason, If the program is halted before the trigger condition is satisfied, the trigger timing is changed. The purpose of this process is to execute instruction part after TCP reaches to its destination point. This prevents earlier trigger because of halt and resume.

This means that halt and resume of program changes trigger timing of Distance Before. Not to change radius of trigger region, set $DISTBF_TTS to 0 (default value: 1).

Example Default configuration
Suppose following program is executed.

```
1: L P[1] 2000mm/sec FINE
2: L P[2] 2000mm/sec CNT100 DB 100.0mm CALL SUB
3: L P[3] 2000mm/sec CNT100
```

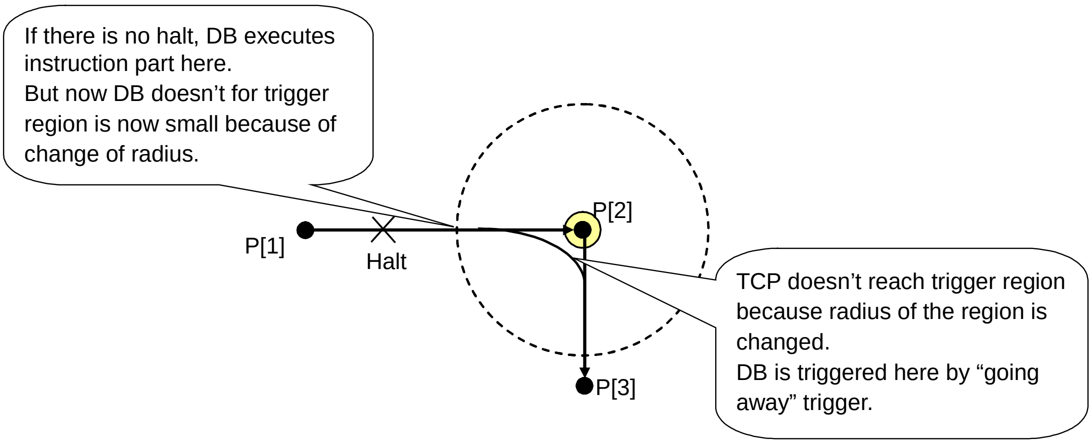


Fig. 9.6.4 (e) Trigger timing after resume of program.

Example Resume with $DISTBF_TTS = 0

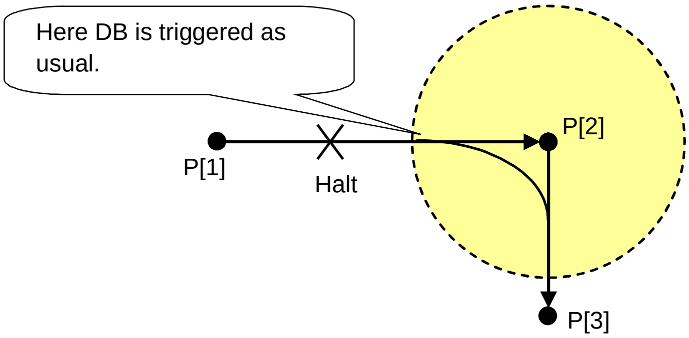


Fig. 9.6.4 (f) $DISTBF_TTS = 0

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
```

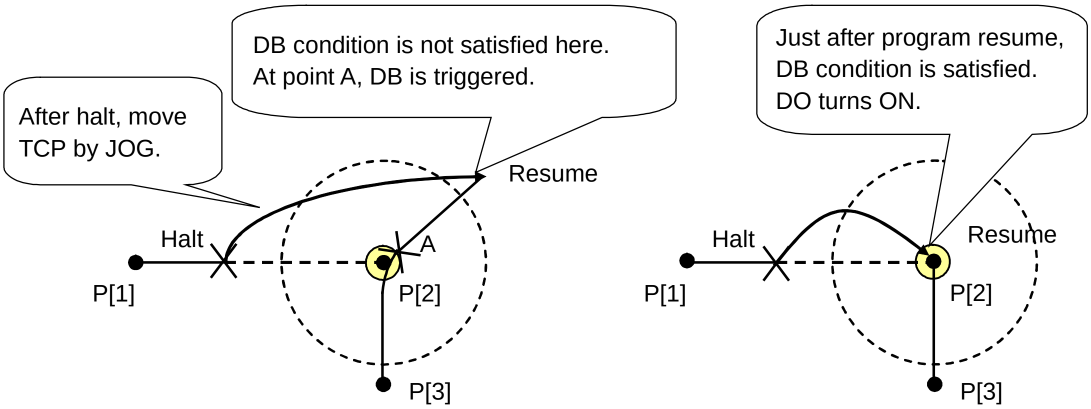


Fig. 9.6.4 (g) Resume after JOG

(ii) $DISTBF_TTS = 0

Radius of trigger region is not changed by halt and resume of program.
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.9.6.4 (h).
If TCP is in trigger region when you resume program, DO[1] turns ON just after resume. (right diagram in Fig. 9.6.4 (h).

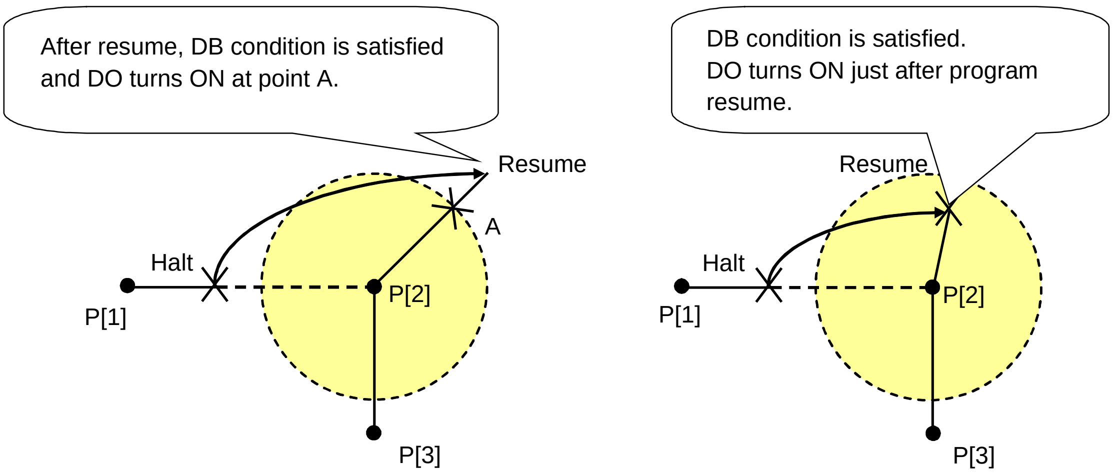


Fig. 9.6.4 (h) Resume after JOG($DISTBF_TTS = 0)

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.

9.6.5 Entering Distance before

1 DB call program

1) Move cursor to motion option area.

```
PNS0001
                                  1/2
   1: J  P[1] 100% FINE
  [End]



                       [CHOICE]
```

2) Press F4, [CHOICE]. List of motion option will be displayed.

```
Motion Modify   1     Motion Modify   2     Motion Modify   3
1 No option           1 Incremental         1 TIME AFTER
2 Wrist Joint         2 Tool_Offset         2 DISTANCE BEFORE
3 ACC                 3 Tool_Offset,PR[     3 PTH
4 Skip,LBL[]          4 Independent EV      4
5 BREAK               5 Simultaneous EV     5
6 Offset/Frames       6 TIME BEFORE         6
7 Offset,PR[ ]        7 Skip,LBL,PR         7
8 --next page--       8 --next page--       8 --next page--
```

3) Select DISTANCE BEFORE. DB is added to program.

```
PNS0001
                                  1/2
   1: J  P[1] 100% FINE DB     0.0mm ...
    :
  [End]






  Enter value
   REGISTER            [CHOICE]
```

4) Input distance value and press the [ENTER] key. The menu to select instruction part will be displayed.

```
PNS0001
                                  1/2
 TIME statement 1 100% FINE DB  100.0mm ...
 1 CALL program
 2 CALL program()
 3 DO[ ]=...
 4 RO[ ]=...
 5 GO[ ]=...
 6 POINT_LOGIC
 7 AO[ ]=...
 8
   1: J  P[1] 100% FINE DB  100.0mm ...
                       [CHOICE]
```

5) To use argument, select CALL program( ). If you don't, select CALL program. Program list will be displayed anyway.

```
PNS0001
                                          1/2
  OPEN_HAND                          100.0mm,
  CLOSE_HAND



   PROGRAM  MACRO  KAREL  INDIRECT  STRINGS  >
```

6) Select program to call.

```
PNS0001
                                          1/2
 Parameter select 1 0% FINE DB  100.0mm,
 1 R[ ]             N_HAND(...)
 2 Constant
 3 String
 4 AR[ ]
 5 <None>
 6 <Insert>
 7 SR[ ]
 8
        :   CALL OPEN_HAND(...)
                             [CHOICE]
```

To specify argument, following procedure is required.

7) Select argument type. Screen displayed below is example to use Constant.

```
PNS0001
                                          1/2
       1: J  P[1] 100% FINE DB  100.0mm,
        :   CALL OPEN_HAND(Constant)
    [End]



                             [CHOICE]
```

8) Input value of argument.

- To use more than 2 arguments, move cursor to ")" and press F4, [CHOICE]. The menu to select argument type is displayed. Teach argument by procedure 7) and 8) described above.
- To delete argument, move cursor to argument you want to delete and press F4. Then select <None>.

To add argument to CALL without argument, following procedure is required.

1 Move cursor to program name.

```
PNS0001
                                             1/2
        1: J  P[1] 100% FINE DB  100.0mm,
         :   CALL A
     [End]
```

2 Press the [PREV] key 2 times. Following submenu will be displayed.

```
TIME statement   1
1 CALL program
2 CALL program()
3 DO[ ]=...
4 RO[ ]=...
5 GO[ ]=...
6 POINT_LOGIC
7 AO[ ]=...
8
```

3 Select CALL program ().

```
PNS0001
                                             1/2
   OPEN_HAND                             100.0mm,
   CLOSE_HAND

 PROGRAM  MACRO  KAREL  INDIRECT  STRINGS  >
```

4 Select program to call and teach argument.

2 DB Signal output

1 Do just same procedure 1-4 for DB CALL program. The menu to select instruction will be displayed.

```
PNS0001
                                             1/2
   TIME statement 1 100% FINE DB  100.0mm ...
   1 CALL program
   2 CALL program()
   3 DO[ ]=...
   4 RO[ ]=...
   5 GO[ ]=...
   6 POINT_LOGIC
   7 AO[ ]=...
   8
        1: J  P[1] 100% FINE DB  100.0mm ...

                        [CHOICE]
```

2 Select signal output instruction.

```
PNS0001
                                         1/2
     1: J  P[1] 100% FINE DB  100.0mm,
      :   DO[...]=...
   [End]




   Enter value
      DIRECT  INDIRECT  [CHOICE]  [LIST]
```

3 Input index and output value just as you do for normal I/O instruction.

```
PNS0001
                                         1/2
     1: J  P[1] 100% FINE DB  100.0mm,
      :   DO[1]=ON
   [End]





                        [CHOICE]
```

3 Point Logic

Please refer to 9.7 POINT LOGIC INSTRUCTION.

4 Finding/Replacing Instructions

- Finding Instructions
  You can find program which is used for DB by “find” on F5 pull-up menu. Select "CALL" and 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 replace subprogram of Distance Before by selecting “CALL” on replace item submenu.
  You can replace signal output of Distance Before by selecting “I/O” on replace item submenu.

9.6.6 Caution and Limitations

- Distance Before cannot be used with TIME BEFORE/AFTER.
- More than 10 motion statement with Distance Before cannot be processed at the same time.
- Distance Before calculates distance between current position and destination point cyclically. Because trigger condition is judged by this cyclical check, actual execution timing of instruction part is different from distance value. Instruction part may be executed inside of trigger region. This means the point where instruction is executed is closer than distance value. Degree of error depends on speed of robot. The slower TCP moves, the more accurate execution timing.
- Distance Before is not recovered by power failure recovery if it was attached to CNT motion statement and power is down when the motion is about to complete.
- Distance Before cannot be used with INC, skip and quick skip in a motion statement.

- Distance Before can be used in the program which has two or more groups. In this case, the group that the group number is least is used to calculate the distance to the destination position.
- Robots that don’t have Cartesian coordination are not supported.
- Position data in matrix form is not supported.
- Integrated axis is not supported.
- FANUC Robot F-200*i* is not supported.
- Line tracking is not supported.
- 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.
- When there is DB/TB that is not yet triggered in previous line, DB/TB does not trigger even when trigger condition is satisfied. (DB/TB always triggers after previous DB/TB triggered.)

9.7 POINT LOGIC INSTRUCTION

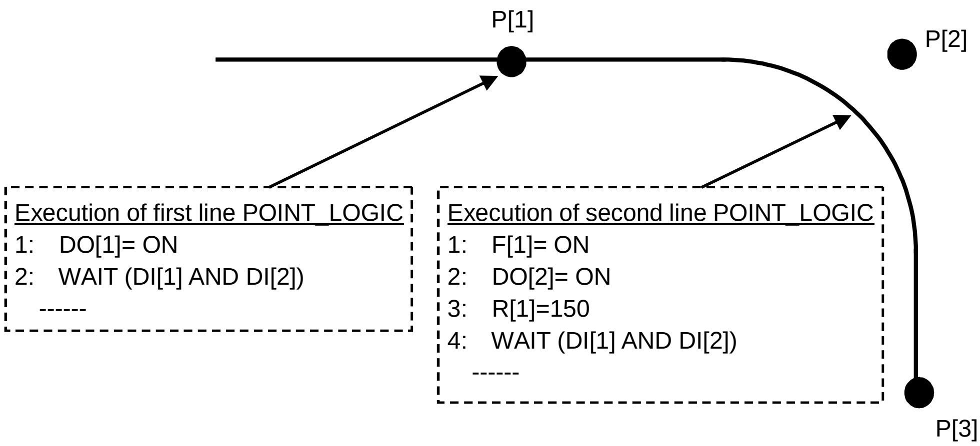
Point Logic instruction is execute multiple instructions at timing executed Timer Before, Distance Before and Time After instructions.
It is not needed creating sub programs to execution multiple instructions by Timer Before instructions.

- Multiple instructions are taught in Point Logic instruction.
- Each Point Logic instruction of lines is independent. Each Point Logic can have different multiple instructions.
- Instructions in Point Logic instruction can be executed at teaching point using Timer Before or Distance Before function.
- Point Logic execution timing can be adjusted with reference to teaching position by changing time value or distance value of the functions.

See ’9.5 Timer Before Function’ and ‘9.6 Distance Before Function’ for more details on Time Before (After) and Distance Before instructions.

```
1:L P[1] 200mm/sec CNT100 DB 100.0mm,
  :    POINT_LOGIC                      Instructions of
                                        First line LOGIC
  1:  DO[1]= ON
  2:  WAIT (DI[1] AND DI[2])
    ------

2: L P[2] 200mm/sec CNT100 DB 100.0mm
  :    POINT_LOGIC                      Instructions of
                                        Second line LOGIC
  1:  F[1]= ON
  2:  DO[2]= ON
  3:  R[1]=150
  4:  WAIT (DI[1] AND DI[2])
    ------
```

Fig. 9.7 (a) POINT_LOGIC Instruction



When distance value of Distance before instruction is 0.0mm, Point Logic instruction is executed at proximate point of teaching position.

Fig. 9.7(b) Distance before function when distance value is 0.0mm

Instruction statement

POINT_LOGIC instruction is used with Time Before (After) or Distance Before function.

```
Motion statement ─┬─ TB  <Time value>      ─┬─  POINT_LOGIC
                  ├─ TA  <Time value>      ─┤
                  └─ DB  <Distance value>  ─┘
```

Fig. 9.7(c) POINT_LOGIC instruction

Example

1: L P[1] 100% FINE TB 1.0sec POINT_LOGIC
2: L P[1] 100% FINE TA 1.0sec POINT_LOGIC
3: L P[1] 100% FINE DB 50.0mm POINT_LOGIC

Procedure 9-11 Entering Point Logic instruction

Step

1 Move the cursor to the space after Timer Before or Distance Before instruction.

```
PROGRAM1
                                            1/2
        1: J  P[1] 100% FINE DB  100.0mm ...
           :
     [End]

|    |    |    | [CHOICE] |    |
```

2 Press F4, [CHOICE]. The menu to select instruction part will be displayed.

```
PROGRAM1
                                             1/2
   TIME statement 1 100% FINE DB  100.0mm ...
   1 CALL program
   2 CALL program()
   3 DO[ ]=...
   4 RO[ ]=...
   5 GO[ ]=...
   6 POINT_LOGIC
   7 AO[ ]=...
   8

        1: J  P[1] 100% FINE DB  100.0mm ...
                          [CHOICE]
```

3 Select POINT_LOGIC. Point Logic instruction is added after Time before or Distance before instruction.

```
PROGRAM1
                                             1/2
        1: J  P[1] 100% FINE DB  100.0mm,
         :   POINT_LOGIC
     [End]

                          [CHOICE]
```

4 Position the cursor on Point Logic instruction or top of line and press the [ENTER] key.
Edit screen of Point Logic instruction is displayed.
If Multi program selection is invalid, edit screen of Point Logic instruction is not used when the program is paused. Abort the program to edit Point Logic instructions.

```
    1:J P[1]
                                             1/1
     [End]

   [ INST ]                            [EDCMD]
```

5 Edit instructions for Point Logic. Operation procedure of this screen is the same as editing of main programs.

```
    1:J P[1]
                                             1/3
        1:   DO[1]=ON
        2:   R[1]=100
     [End]

   [ INST ]                            [EDCMD]
```

6 Press the [PREV] key when editing of Point Logic statement is finished. Point Logic view screen is displayed. Press the [ENTER] key to edit main program or other Point Logic instructions. Edit screen is displayed.

Point Logic View function

Point Logic view screen can be used, when Point Logic instructions exist in programs.
Instructions in Point Logic instruction are uncompressed and displayed in view screen. Main program and instructions in Point Logic instruction can be viewed on the same view screen.

Point Logic view screen is displayed when the [PREV] key is pushed in Edit screen.
When Point Logic instruction does not exist in the program, view screen cannot be displayed.

Edit screen is displayed when ENTER or F2, EDIT key is pushed in View screen.
When the cursor positions on line of main program, Edit screen for main program is displayed.
When the cursor positions on line of Point Logic instructions, Edit screen for Point Logic instruction is displayed.

In Point Logic view screen, the program cannot be edited. Move to Edit screen by the [ENTER] key and edit the program.

When the program is paused while executing Point Logic instruction, the cursor cannot move to line of main program.
Abort the program or execute backward by [SHIFT] key and [BWD] key to move back cursor to line of main program.

If 'Multi program selection' is invalid, the cursor is not stopped in line of Point Logic instructions by manual when the program is paused in main program. The cursor is stopped only line of main program.

Edit screen:

```
PROGRAM1
                                          1/4
      1: L P[1] 200mm/sec FINE DB  100.0mm,
       :   POINT_LOGIC
      2: L P[2] 200mm/sec FINE DB  100.0mm,
       :   POINT_LOGIC
      3: L P[3] 200mm/sec FINE DB  100.0mm,
       :   POINT_LOGIC
   [End]

   [ INST ]                         [EDCMD]   >
```

PREV key

View screen: ENTER key

```
PROGRAM1
      Viewer                              1/4
      1: L P[1] 200mm/sec FINE DB  100.0mm,
       :    POINT_LOGIC
       1:    DO[1]=ON
       2:    WAIT (DI[1] AND DI[2])
        ------
      2: L P[2] 200mm/sec FINE DB  100.0mm,
       :    POINT_LOGIC
       1:    F[1]=ON
       2:    DO[2]=ON
       3:    R[1]=150
       4:    WAIT (DI[1] AND DI[2])
        ------

             EDIT
```

Fig.9.7(d) Point Logic view screen (Main program)

Edit screen:

```
   2:L P[2]
                                     1/5
       1:   F1[1]=ON
       2:   DO[2]=ON
       3:   R[1]=150
       4:   WAIT (DI[1] AND DI[2])
    [End]

   [ INST ]                       [EDCMD]
```

PREV key

View screen: ENTER key

```
PROGRAM1
      Viewer                         1/4
        1:    DO[1]=ON
        2:    WAIT (DI[1] AND DI[2])
         ------
       2: L P[2] 200mm/sec FINE DB  100.0mm,
        :    POINT_LOGIC
        1:    F[1]=ON
        2:    DO[2]=ON
        3:    R[1]=150
        4:    WAIT (DI[1] AND DI[2])
         ------
       3: L P[3] 200mm/sec FINE DB  100.0mm,
        :    POINT_LOGIC

            EDIT
```

Fig.9.7(e) Point Logic view instruction（in point logic statements）

Precautions

Motion instructions can not added in Point Logic instructions.
When line with Point Logic instruction is copied, instructions in Point Logic instruction are also copied.
When Point Logic instruction is overwritten by Call instructions or other instruction, instructions in Point Logic instruction are deleted.
When line with Point Logic instruction is deleted, instructions in Point Logic Instruction are also deleted.

9.8 SIGNAL OUTPUT MOTION OPTION

Signal output motion option is the function to teach multiple signal output in TIME BEFORE instruction. Timing of signal output is same as that of 0.0sec specified Time before instruction.

- Multiple signal output can be taught on one motion instruction.
- Signal output motion option is easier to teach than normal time before instruction.

Ex)
1: J P[1] 100% FINE
2: L P[2] 1000mm/sec CNT100 DO[1]=ON DO[2]=ON
3: L P[3] 1000mm/sec FINE DO[1]=OFF DO[2]=OFF

Instruction statement

Following signal can be used. Maximum 5 signal output can be taught on one motion instruction.

DO[] RO[]	=	ON OFF R[] pulse
GO[] AO[]	=	Constant R[] AR[ ]

Procedure 9-10 Entering the SIGNAL OUTPUT MOTION OPTION

Step

1 Move cursor to motion option area.

```
PNS0001
                                  1        1/2
        1: J  P[1] 100% FINE
     [End]

                                [CHOICE]
```

2 Press function key F4, [CHOICE]. A list of motion option instructions will be displayed.

```
Motion Modify   1     Motion Modify   2      Motion Modify   3
1 No option           1 Incremental          1 TIME AFTER
2 Wrist Joint         2 Tool_Offset          2 DISTANCE BEFORE
3 ACC                 3 Tool_Offset,PR[      3 DO[]=...
4 Skip,LBL[]          4 Independent EV       4 RO[]=...
5 BREAK               5 Simultaneous EV      5 GO[]=...
6 Offset/Frames       6 TIME BEFORE          6 AO[]=...
7 Offset,PR[ ]        7 Skip,LBL,PR          7 PTH
8 --next page--       8 --next page--        8 --next page--
```

3 Select signal type. If you select DO[], following screen will be displayed.

```
PNS0001
                                          1/2
        1: J  P[1] 100% FINE DO[...]=...
     [End]




    Enter value
  DIRECT  INDIRECT  [CHOICE]  [LIST]  DIRECT
```

4 Input index and output value just as you do for normal I/O instruction.

```
PNS0001
                                          1/2
     1: J  P[1] 100% FINE DO[1]=ON
  [End]

                        [CHOICE]
```

Precautions

Signal output motion option cannot be used with TIME BEFORE/AFTER.
Signal output motion option is internally equivalent with time before instruction.
So signal output motion option has same restrictions that time before instruction has.
About restrictions of time before instruction, please refer to [9.5 TIME BEFORE FUNCTION].

9.9 CONDITION MONITOR 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.
  Example:
  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 RI[2]=Off, CALL STP_RBT
  *1 *2
  This condition program states that when RI[2] turns off, STP_RBT is to be called.
  *1 Describe the desired monitoring condition by following instruction WHEN. The types of monitoring condition are explained in the WHEN section.
  *2 Specify the program to be executed if the condition described in *1 is satisfied. The action program can be created and named in the same way as a normal program.

- Action program
  Called if the condition is satisfied. The same instructions as those used in normal programs can be used.
  Program example:
  1:RO[2]=ON ! Notification to a peripheral device

```
2:R[8]=R[8]+1 ! Drop count
3:UALM[1]     ! Alarm and robot stop
```

The alarm message corresponding to user alarm [1] is set to the system variables as follows.
$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
     :
     :                  Handling Operation   Condition monitoring
     :
     :
 8: J  P[7] 100% FINE
 9:   MONITOR END WRK_FALL
10:   Open hand
```

Workpiece drop. condition (condition program)

```
 1:   WHEN RI[2]=Off, CALL STP_RBT
```

Robot stop. TP (action program)

```
 1:   RO[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 condition monitoring within a separate program.
Monitoring starts with an instruction (monitor start instruction) in the program. Monitoring terminates with a monitor stop instruction or program termination.
The program monitor can be switched between two settings: setting 1 in which the monitor stops when the program stops temporarily, and setting 2 in which the monitor continues monitoring.

> **NOTE**
> Settings 1 and 2 cannot be used at the same time.

- System monitor

This type of monitor does not depend on the execution state of the program. It is suitable for monitoring the state of the entire system.
The monitor is started and stopped from the state screen. It cannot be operated with instructions in the program.
The system monitor can be switched between two settings: setting 1 in which the monitor stops after a cold start, and setting 2 in which the monitor continues monitoring.

> **NOTE**
> The program monitor and the system monitor can be used at the same time.

The monitors can be switched between the settings using the following system variables:
Following system valuables can be changed by system valuable menu under controlled start mode.

$TPP_MON.$LOCAL_MT = 1 or 3	Switches the program monitor to setting 1 (default).
$TPP_MON.$LOCAL_MT = 2 or 4	Switches the program monitor to setting 2 (Same specification as that for KAREL).
$TPP_MON.$GLOBAL_MT= 0	Enables the system monitor (default).
$TPP_MON.$GLOBAL_MT= 1	Switches the system monitor to setting 1.
$TPP_MON.$GLOBAL_MT= 2	Switches the system monitor to setting 2.

Notice) About $TPP_MON.$LOCAL_MT = 3, 4

The default value of the system variable $TPP_MON.$LOCAL_MT is 3.

In case $TPP_MON.$LOCAL_MT = 3, 4
Action program with motion group cannot be used. (Alarm will occur when program is executed.)

This is to prevent unintentional motion caused by a wrong action program with motion instruction.
When the alarm occur, please specify the operation group in action program as [*,*,*,*,*,*,*,*] .
When the operation group in the action program cannot be set as [*,*,*,*,*,*,*,*], please use traditional setting. (Use $TPP_MON.$LOCAL_MT = 1 or 2.)

Monitor state transition

The states of the monitors assumed when each operation is performed are listed in the table below:

Operation	Program monitor		System monitor	
	Setting 1	Setting 2	Setting 1	Setting 2
MONITOR instruction	A	A	E	E
RESTART (state screen)	B	B		
START (state screen)			A	A
Program Stop	C	E *1)	E	E
Program End/Enforced End	D	D	E	E
MONITOR END	D	D	E	E
PAUSE (state screen)	C	C	D	D
END (state screen)	D	D		
RESUME	B	B	E	E
Power failure handling Power off with monitoring state	C	E	E	E
Power failure handling Power off without monitoring state	E	E	E	E
START (COLD)	D	D	D	E
CONTROLLED START	D	D	D	D
Other operation	E	E	E	E

Meanings of symbols

A : Condition monitoring is started. / B : Condition monitoring is restarted if it is stopped.
C : Condition monitoring is stopped. / D : Condition monitoring is deleted. (Cannot be restarted)
E : The state of condition monitoring does not change due to the operation.
*1 : Monitoring continues, but the action program will pause even if the conditions are satisfied.
Blank : The operation is impossible.

Operation-by-operation description

Operation	State
MONITOR instruction	When a monitor start instruction in the program is executed, monitoring with the specified program monitor starts.
RESTART (state screen)	When function key RESTART is pressed on the Program monitor screen of the state screen, monitoring with the program monitor specified with the cursor restarts.
START (state screen)	When function key START is pressed on the System monitor screen of the state screen, monitoring with the system monitor specified with the cursor starts.
Program Stop	When the temporary stop key is pressed or if the program stops temporarily due to the occurrence of an alarm, condition 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 cycled. ● Condition monitoring stops if the program monitor is set to setting 1. ● Condition monitoring continues if the program monitor is set to setting 2. (The program stops temporarily, but condition monitoring is performed.) ● The system monitor continues condition monitoring. If the monitor is stopped, it remains in the stopped state when the power is cycled.
Cold start	If power failure handling is disabled and the power is cycled, 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

Condition 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 Cond, can specify condition instructions only.

- WHEN <conditional-expression>, CALL <program-name>

The following conditions can be used:

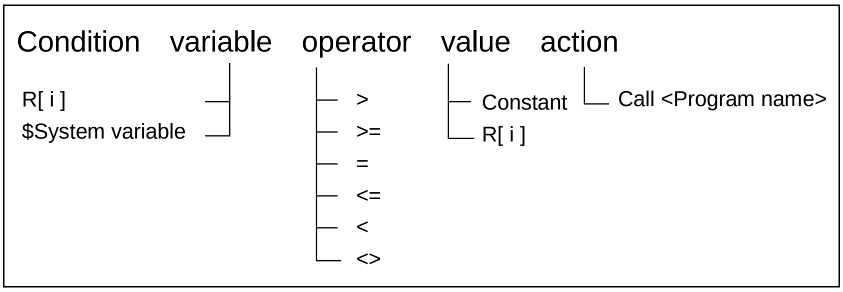


Fig. 9.9 (a) Register/system variable condition compare instruction

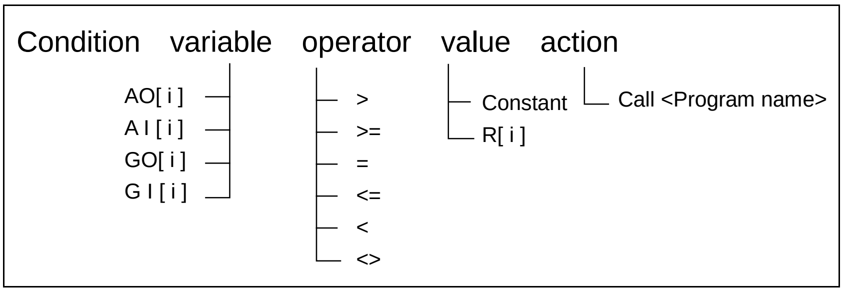


Fig. 9.9 (b) I/O condition compare instruction 1

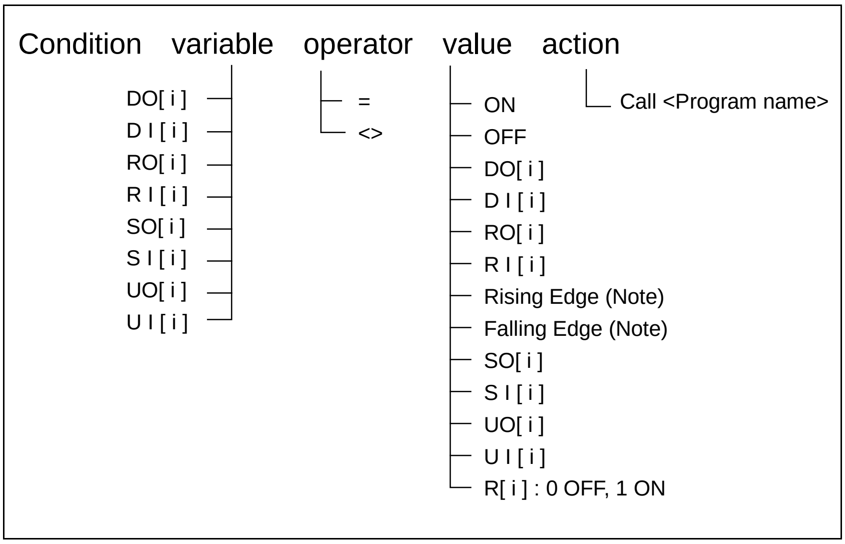


Fig. 9.9 (c) I/O condition wait instruction 2

NOTE

Falling edge:
The falling edge of a signal is regarded as being a detection condition. The condition is not satisfied when the signal remains off. The detection condition is satisfied when the signal changes from the on state to the off state.
Rising edge:
The rising edge of a signal is regarded to be a detection condition. The condition is not satisfied when the signal remains on. The detection condition is satisfied when the signal changes from the off to the on state.

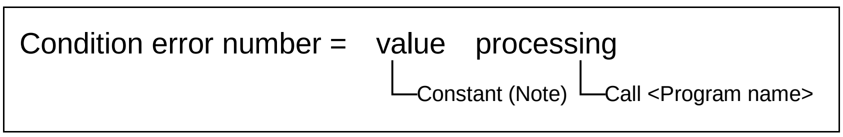


Fig. 9.9 (d) Error condition compare instruction

> **NOTE**
> An error number is specified with an alarm ID followed by an alarm number.
> Error number = aabbb
> where aa = alarm ID
> bbb = alarm number
> For an explanation of alarm IDs and numbers, refer to the “FANUC Robot series R-30*i*B/R-30*i*B Mate CONTROLLER OPERATOR’S MANUAL (Alarm Code List)” (B-83284EN-1).
> (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 will be displayed:

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 displayed menu.

> **NOTE**
> At this time, the operation group is automatically set as [*,*,*,*,*,*,*,*].
> A condition program requires no operation group.

Condition monitoring screen

The state of condition 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 (Note) 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
                                        1/3
          CH Program   Status   Program
      1   WORKDROP      Running  SAMPLE
      2   WELDTIME      Paused   SAMPLE
      3   NOWORK_L      Paused   SAMPLE2

   [ TYPE ]  SYSTEM  RESTART  PAUSE   END
```

Table 9.9 (a) Items and function keys on the program monitor screen

Item	Description
CH Prog.	Condition program name
Status	State of the program, either being executed or stopped
Program	Name of the parent program of the program that started the program monitor
F2 SYSTEM	Switches the screen to the system monitor screen. If the system monitor is disabled ($TPP_MON.$GLOBAL_MT=0), this key is not effective.
F3 RESTART	When pressed, this key restarts the stopped monitor.
F4 PAUSE	Stops the monitor.
F5 END	Terminates the monitor. The terminated monitor is cleared from the screen.

System monitor screen
All condition programs are displayed. System monitors can be started and stopped.

```
System Monitor
                                        1/3
          CH Program                Status
      1   WORKDROP                  Running
      2   WELDTIME
      3   NOWORK_L
      4   VRFY_HND

   [ TYPE ]  PROGRAM          START   END
```

Table 9.9 (b) Items and function keys on the system monitor screen

Item	Description
CH.Prog.	Condition program name
Status	State of the program, either being executed or not started (blank)
F2 PROGRAM	Switches the screen to the program monitor screen.
F3 START	Starts the system monitor.
F5 END	Stops the monitor. In the “State” column, a blank is displayed for the stopped monitor.

Notes/restrictions

If multiple condition instructions are specified in a condition program, multiple monitors are started at the same time.

```
1:WHEN (conditional-expression1),  CALL (Program name1)
2:WHEN (conditional-expression2),  CALL (Program name2)
3:WHEN (conditional-expression3),  CALL (Program name3)
```

If, before one monitor start instruction terminates, another monitor start instruction is executed, both monitors are executed at the same time.
If the condition program names specified in the monitor start instructions are the same, the first condition program is overwritten by the second.
The program monitor stops condition monitoring under the following conditions:

- The MONITOR END instruction is executed.
- The program terminates.
- The program stops temporarily. (Condition 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-expression 1) and (conditional-expression 2) .........  and (conditional-expression 5)
condition (conditional-expression n)or (conditional-expression m)  .........  or (conditional-expression l)     Up to
                    :                                 :                                  :                     ten
condition (conditional-expression o) and (conditional-expression p).........  and (conditional-expression q)
```

While the program is being executed or while it is stopped, the condition statements (condition program) cannot be edited.
In the action program for a system monitor and a program monitor, an operation group cannot be specified.
In the action program, the operation group must be specified as [*,*,*,*,*,*,*,*].
If the condition is satisfied, the condition program enters the END state. If condition monitoring is to continue, specify a monitor start instruction in the program. Clear the monitoring condition beforehand.
Example

```
MON1. TP
  1:WHEN R[1]=1 CALL ACT1
ACT1.TP
  1:R[1]=0          <-----------  Clear the condition
  2:
  3:(Action)
       :
  9:MONITOR MON1    <-----------Start the monitoring condition again
```

If there is no line on which the condition on line 1 is dropped, the condition is immediately satisfied on the monitor start instruction on line 9, causing a MEMO-065 error.

The condition program cannot be executed directly.

9.10 COLLISION DETECTION FOR AUXILIARY AXIS

9.10.1 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 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.

9.10.2 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.

9.10.3 Initial Setting

1 Setup auxiliary axis (Gear ratio, acceleration time, and etc.) normally.
2 Turn on the controller power.
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.

9.10.4 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.
On this occasion, note that this program must not be paused because the disturbance torque displayed in the screen explained below will be cleared at just restarting it.
3 Measure the max. / min. disturbance torque on STATUS/AXIS/ DISTURB screen after running the program.

```
STATUS Axis

                                 GRP[ 1 ]
       Disturbance Torque (A)
       Curr./ Max.(Allowed)/Min.(Allowed)
  J1 :   0.0   20.0(  40.0)  -19.0( -40.0)
  J2 :   0.0   19.0(  40.0)  -20.0( -40.0)
  J3 :   0.0   22.0(  40.0)  -10.0( -40.0)
  J4 :   0.0   12.0(  20.0)   -5.0( -20.0)
  J5 :   0.0   10.0(  20.0)  -11.0( -20.0)
  J6 :   0.0    8.0(  20.0)   -4.0( -20.0)
  J7 :   0.0   24.0(  56.0)  -30.0( -56.0)

 [ TYPE ] | 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 programs 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.

```
STATUS Axis

                                 GRP[ 1 ]
       Disturbance Torque (A)
       Curr./ Max.(Allowed)/Min.(Allowed)
  J1 :   0.0   20.0(  40.0)  -19.0( -40.0)
  J2 :   0.0   19.0(  40.0)  -20.0( -40.0)
  J3 :   0.0   22.0(  40.0)  -10.0( -40.0)
  J4 :   0.0   12.0(  20.0)   -5.0( -20.0)
  J5 :   0.0   10.0(  20.0)  -11.0( -20.0)
  J6 :   0.0    8.0(  20.0)   -4.0( -20.0)
  J7 :   0.0   24.0(  24.0)  -30.0( -30.0)

 [ TYPE ] | MONITOR | TRACKING | DISTURB | [UTIL ] | >
```

NOTE

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 x Max. current of amp.

Lower Limit = Min. allowed value - 0.3 x Max. current of amp.

SRVO-050 Collision Detect alarm (G:x,A:x)

Part of 0.3 × Max. current of amp. is the margin to prevent the miss-detection.
For example in above screen with 40A amplifier,

Upper Limit = 24.0 + 0.3 x 40 = 36 A

Lower Limit = -30.0 - 0.3 x 40 = -42 A

5 Run the programs again with above disturbance allowed setting, and confirm that there is no miss-detection.

9.11 PASSWORD FUNCTION

9.11.1 Overview of the Password Function

A password is a combination of up to 12 characters, digits, and symbols. It is used to restrict various operations and screen accesses by operators. Password protection is disabled unless an install user is defined.
There are eight password levels, and the possible operations and displayable screens differ depending on the level. Levels 3 to 7 are user-definable. For an overview of password authentication for the four levels, see Table 9.11.1 Password levels.

Table 9.11.1 Password levels

Level	Operation
Install	Enables the user to allocate **user** names, **passwords**, and **levels**. Enables the user to clear user names and passwords. Enables the user to enable or disable the password log. Allows the user to set the number of password users in the system. Enables the user to execute all operations that settings users, programs, and operators can executed. **Caution** Only **one** install user can be set.
Setup	Enables the user to execute operations generally performed for system setting.
Program	Enables the user to execute operations more advanced than those that production users can execute.
Operator	Enables the user to execute basic operations.
User-Defined	Enables the user to execute basic operations unless otherwise defined in the password configuration file.

> ⚠ **CAUTION**
> In a standard system, the user is set to the operator level when the controller is turned ON.

> ⚠ **WARNING**
> There are some functions that cannot be executed without the password of the install user. If the password of the install user is forgotten, contact the FANUC service representative.

Password operations

To use passwords, set an install user first. To set an install user, it is necessary to assign an install user name and a password first and then log in. Once logged in, the install user can assign the user name, level, and password for each of other users.

> ⚠ **CAUTION**
> The password function cannot be used unless an install user name and a password are assigned.

After the install user has assigned a user name, password level, and password to a user, this user must log in to work at the assigned level. The user selects his user name and input his password to log in. Only one user can log in to one device at a time. The term device, as used here, refers to a teach pendant, CRT/keyboard, KCL, and so on.

When the user has completed work, he logs out. If he does not log out, a timeout occurs when the time specified for user timeout has elapsed. When the time specified for user timeout has elapsed or the user logs out, the system returns to the operator level, so that another user can log in. If a user forgets to log out, there are cases in which the user is forced to log out by another user.
If the install user enables log recording on the password setting screen, the password log can be displayed on the alarm screen. In the password log, important information about data changes (descriptions of changes made by users and the times the changes were made) is displayed. For details, see Procedure 9-22 "Displaying the password log".
For information on the assignment of a user name, password level, and password by the install user, see the Subsection 9.11.2, "Password Operations by the Install User". For information on each of operator, program, and setup users, see the Subsection 9.11.4, "Password Operations by Program Users and Setup Users".

9.11.2 Password Operations by the Install User

The install user is required to perform the operations below.

- Assign an install user name and a password (Procedure 9-12 "Assignment a user name and a standard password for each password level")
- Assign the user names, levels, and passwords for all other users (Procedure 9-12 "Assignment a user name and a standard password for each password level")
- Enable or disable the password log and display it if enabled (Procedures 9-21 "Enabling the password log" and Procedures 9-22 "Displaying the password log" in the "Password Log" section)

For information on setting a password level, see Procedure 9-12 "Assignment a user name and a standard password for each password level".

Procedure 9-12 Assignment a user name and a standard password for each password level

Step

1 Press [MENU] key.
2 Select SETUP.
3 Press F1, [TYPE].
4 Select Passwords. The screen below will be displayed.

```
SETUP Passwords

         Password are inactive.
         Press LOGIN to activate.

   [ TYPE ]   LOGIN
```

5 Press F2, LOGIN. The screen below will be displayed.

```
SETUP Passwords
                                         1/1
         USERNAME   PWD LEVEL       TIME(min)
      1             *   INSTALL            0

    Press ENTER to add INSTALL user.

   [ TYPE ]   LOGIN    LOGOUT              HELP    >
```

> ⚠ **CAUTION**
> It is necessary to set an install user name and a password first.

6 Use the steps below to assign an install user name and a password.
- a Press the [ENTER] key.
- b Use the arrow keys and function keys as appropriate to input an install user name. When finished, press the [ENTER] key. The screen below will be displayed.

```
                                       2/3



  --Set password for BOB                      --

     Old password: '              '
     New password: '           Alpha input  1
     Verification: '            Upper Case
                                Lower Case
                                Punctuation
 Old Value:                     Options

 | ABCDEF | GHIJKL | MNOPQR | STUVWX | YZ_@* |
```

> ⚠ **CAUTION**
> A password must consist of at least three characters.

> ⚠ **WARNING**
> Take a note of the install password. There are some functions that cannot be executed without the install password. If you forget the install password, contact the FANUC service representative.

- c Input a new password, and press the [ENTER] key.
- d Input the new password again to confirm that the previous input is correct and then press the [ENTER] key. The screen below will be displayed.

```
  Would you like to be logged in?[YES]

 |        |        |        |  YES   |   NO   |
```

7 To log in, press F4, YES. Otherwise, press F5, NO.

> ⚠ **CAUTION**
> Unless logged in as an install user, the user cannot set other users.

If F4, YES is pressed, the screen below will be displayed.

```
SETUP Passwords
                                          1/4
        Current user:        BOB
        Current level:       INSTALL
      1 Default user timeout:       15 min

      2 Timeout occurs in:          15 min
      3 Log events:             DISABLE
      4 Number of users:           10

   [ TYPE ]   LOGIN   LOGOUT   PASSWORD   HELP   >
```

8 If F2, LOGIN is pressed, the user list screen will be displayed as follows.

```
SETUP Passwords
                                            1/10
         USERNAME     PWD LEVE       TIME(min)
      1@BOB             -  INSTALL         15
      2                 *  ************     0
      3                 *  ************     0
      4                 *  ************     0
      5                 *  ************     0
      6                 *  ************     0
      7                 *  ************     0
      8                 *  ************     0
      9                 *  ************     0
     10                 *  ************     0

   [ TYPE ]   LOGIN   LOGOUT           HELP   >
```

⚠ CAUTION
The @, which appears when the user logs in, is a symbol indicating a login user name.

9 Use the steps below to assign the name, password, and level for a second user.
  a Move the cursor to the second user name input item, press the [ENTER] key, and then use the function keys as appropriate to input a name.
  b Move the cursor to the PWD field, press the [ENTER] key and then use the function keys as appropriate to input a password.
  c Move the cursor to the LEVEL field, press F4, [CHOICE] and then select a level.
  d Move the cursor to the TIME field and input a user timeout value. The user timeout value may be in the range of 0 to 10080 minutes (7 days).

⚠ CAUTION
If the user logs in with a user timeout value of 0, no timeout is performed. A value greater than 0 indicates the length of time (in minutes) of the pause until the system performs a timeout.

  e For other users that are to be permitted to access the system, perform a through d of step 9.

10 To clear an input user name and password, press the [NEXT] key and then F2, CLEAR.
11 To clear the user names and passwords of all users other than the install user, press the [NEXT] key and then F3, CLR_ALL.
12 To change the number of users in the system, use the steps below.

> ⚠ **WARNING**
> If the number of users in the system is changed to a number less than the number of assigned users, some users may be deleted from the system.

a Press the [PREV] key to display the password setting screen, which appeared first.
b Move the cursor to the number of users field and input a value. The allowable number of users is in the range of 10 to 100.

If increasing the number of users
c Input a new number of users and press the [ENTER] key. The screen below will be displayed.

```
SETUP Passwords
                                          4/4
        Current user:        BOB
        Current level:       INSTALL
      1 Default user timeout:       15 min

      2 Timeout occurs in:          15 min
      3 Log events:            DISABLE
      4 Number of users:            15



     Perform COLD start to see new users.
   [ TYPE ]   LOGIN   LOGOUT   PASSWRD   HELP   >
```

If the number of users is to be decreased, the prompt below will be displayed.

```
SETUP Passwords
                                          4/4
        Current user:        BOB
        Current level:       INSTALL
      1 Default user timeout:       15 min

      2 Timeout occurs in:          15 min
      3 Log events:            DISABLE
      4 Number of users:            12



    Reconfiguring.  DELETE users?[NO]
                                   YES      NO
```

To delete users, press F4, YES. To cancel the operation, press F5, NO.

d Cycle power of the controller to enable the new list of users.

13 Press F3, LOGOUT to log out.

> ⚠ **CAUTION**
> If the user timeout time elapsed or the user logs out, the device in use returns to the operator level. If the controller is turned off, all devices return to the operator level.

9.11.3 Disabling the Password Function

In a system in which a password is set, it is possible to disable the password function. Disabling the password function causes the install user to be cleared. No other users are cleared. Only install level user can disable the password function but the others can also disable the password function by receiving password release key from FANUC service in case system administrator is changed or password is lost or etc. For information on disabling the password function, see Procedure 9-13 "Disabling the password function" in case Install user operates, see Procedure 9-14 "Forced releasing the password (directly input password key) " or Procedure 9-15 "Forced releasing the password (loading password key file) " in case non-Install user operates. In Procedure 9-15 "Forced releasing the password (loading password key file) ", it is possible to disable the password function of multiple robots by only one password release key file.

Procedure 9-13 Disabling the password function

Condition

- The user is logged in as an install user.

Step

1 Press [MENU] key.
2 Select SETUP.
3 Press F1, [TYPE].
4 Select Passwords. The screen below will be displayed.

```
SETUP Passwords
                                             1/4
        Current user:        JOE
        Current level:       INSTALL
      1 Default user timeout:      15 min

      2 Timeout occurs in:         15 min
      3 Log events:           DISABLE
      4 Number of users:           10

   [ TYPE ]   LOGIN   LOGOUT   PASSWRD   HELP   >
```

5 Press NEXT ">".
6 Press F3, DISABLE. The screen below will be displayed.

```
SETUP Passwords
                                             1/4
        Current user:        JOE
        Current level:       INSTALL
      1 Default user timeout:      15 min

      2 Timeout occurs in:         15 min
      3 Log events:           DISABLE
      4 Number of users:           12

    Disable passwords?[NO]
                                YES      NO
```

> ⚠ **CAUTION**
> Disabling the password function causes the install user to be cleared. No other users are cleared. Disabling the password function requires the install level.

7 To disable the password function, press F4, YES. To cancel disabling the password function, press F5, NO.

Procedure 9-14 Forced releasing the password (directly input password key)

Condition

- The user is not logged in as an install user.
- You are system administrator.

Step

1 Press [MENU] key.
2 Select SETUP.
3 Press F1, TYPE.
4 Select Passwords. The following screen will be displayed.

```
SETUP Passwords
                                        1/4
        Current user:        Nobody
        Current level:       OPERATOR
      1 Default user timeout:      0 min

      2 Timeout occurs in:          0 min
      3 Log events:           DISABLE
      4 Number of users:          10

  [ TYPE ]  LOGIN   LOGOUT  PASSWRD   HELP   >
```

5 Press NEXT ">".
6 Press F3, DISABLE. The following screen will be displayed.

```
Password Release
                                        1/1
     Release key required if Password is
     To be disabled by non-Install user.
     Install user please log in first.

        Password Release ID:   823972321
      1 Password Release key:

   PREV   OUTPUT   LOAD   RELEASE
```

7 Write down Password Release ID.
8 Convey what you want to release password protection to your local FANUC representative with Password Release ID by telephone.
9 Write down Password Release key what is conveyed by your local FANUC representative .

NOTE
1 Password Release ID is 9 digit. Password Release key is 12 characters.
2 There are situations that Password Release key isn't conveyed immediately and one that you are checked whether system administrator or not. Please accept these situations ahead of time.

10 Input Password Release key in PASSWORD RELEASE menu and press F4, RELEASE key. The prompt message below will be displayed.

```
Disable passwords?[NO]

|   |   |   | YES | NO |  |
```

⚠ CAUTION
Disabling the password function causes the install user to be cleared. No other users are cleared. Disabling the password function requires the install level.

11 To disable the password function, press F4, YES. To cancel disabling the password function, press F5, NO.

Procedure 9-15 Forced releasing the password (loading password key file)

Condition

- The user is not logged in as an install user.
- You are system administrator.

Step

1 Press [MENU] key
2 Select SETUP.
3 Press F1, TYPE.
4 Select Passwords. The screen below will be displayed.

```
SETUP Passwords
                                              1/4
        Current user:        Nobody
        Current level:       OPERATOR
      1 Default user timeout:      0 min

      2 Timeout occurs in:         0 min
      3 Log events:            DISABLE
      4 Number of users:           10

 [ TYPE ]  LOGIN  LOGOUT  PASSWRD  HELP  >
```

5 Press NEXT ">".
6 Press F3, DISABLE. The screen below will be displayed.

```
Password Release
                                              1/1
     Release key required if Password is
     To be disabled by non-Install user.
     Install user please log in first.

        Password Release ID:   823972321
      1 Password Release key:

     PREV    OUTPUT    LOAD    RELEASE
```

7 Press F2, OUTPUT. The prompt message below will be displayed.

```
   Select device for outputting release ID

             CANCEL    MC:     UD1:     UT1:
```

8 Select device that release ID file (pwdrlsid.dat) should be output. It is possible to output multi release IDs to one file. The prompt message below will be displayed in case there has been release ID file (pwdrlsid.dat) already. Press F3, APPEND if you want to append release ID.

```
   There has been output file in MC:\

                      APPEND            CANCEL
```

9 The prompt message below will be displayed in case release ID is output.

```
   ID was output to MC:\pwdrlsid.dat

     PREV    OUTPUT    LOAD    RELEASE
```

10 Convey what you want to release password protection to FANUC service with attaching Password Release ID file by E-mail.

11 Password release key file (pwdrlsky.dat) is sent by FANUC service. Put this file on root directory of external memory device and connect this device to robot controller.

> **NOTE**
> There are situations that Password Release key isn’t conveyed immediately and one that you are checked whether system administrator or not. Please accept these situations ahead of time.

12 Press F3, LOAD in PASSWORD RELEASE menu. The prompt message below will be displayed.

```
   Select device for loading release key

             CANCEL    MC:     UD1:     UT1:
```

13 Select device that release key file (pwdrlsky.dat) is put on root directory. Release key is loaded as following display.

```
Password Release
                                              1/1
     Release key required if Password is
     To be disabled by non-Install user.
     Install user please log in first.

        Password Release ID:    823972321
      1 Password Release key:  ASFA2F4NF3N9

     PREV    OUTPUT    LOAD    RELEASE
```

14 Press F4, RELEASE key. The prompt message below will be displayed.

```
   Disable passwords?[NO]

                                 YES      NO
```

> ⚠ **CAUTION**
> Disabling the password function causes the install user to be cleared. No other users are cleared. Disabling the password function requires the install level.

15 To disable the password function, press F4, YES. To cancel disabling the password function, press F5, NO.

9.11.4 Password Operations by Program Users and Setup Users

Program users and setup users can perform the operations below.

- Log in (Procedure 9-16 "Logging in")
- Log out (Procedure 9-17 "Logging out")
- Change their own passwords (Procedure 9-18 "Changing a password")
- Displaying the password log (Procedure 9-22 "Displaying the password log" in the "Password Log" section)

Table 9.11.4 (a) Items on the password setting screen - Login screen

Item	Explanation
Current user	This item indicates the user who is currently logged in.
Current level	This item indicates the current system protection level.
Default user timeout	This item indicates the time after which the system automatically returns to the operator level if the user does not execute any operations after logged in.
Timeout occurs in	This item indicates the current remaining time after which the system returns to the operator level.
Log events	This item indicates whether to record operations in the log.
Number of users	This item indicates the maximum allowable number of users in the system.

Table 9.11.4 (b) Items on the password setting screen - User screen

Item	Explanation
USERNAME	This column indicates the names set as password users.
PWD	This column indicates whether passwords are set for users.
LEVEL	This column indicates password protection levels for users.
TIME	This column indicates timeout values for users. A timeout value is the length of time that the login status of a user is retained for the specified password level if he does not perform any operations.
INSTALL user	Assign the user name and password for an install user, so that a user can *log in* as an install user. Once logged in, the install user can *assign* the user name, level, and password for each of other users.

Procedure 9-16 Logging in

Condition

- A password has been set. (See the section, "Password Operations by the Install User".)
- No other user is logged in. Only one user can log in at a time.

> ⚠ **CAUTION**
> A user who does not know his user name or password should contact the install user.

Step-a (In case that no user logs in)

1 Display the password setting screen.
- If the full menu is used, the procedure is as follows:
  a Press [MENU] key.
  b Select SETUP.
  c Press F1, [TYPE].
  d Select Passwords.
- If the quick menu is used, the procedure is as follows:
  a Press [MENU] key.
  b Select SETUP PASSWORDS.

2 Press F2, LOGIN. The screen below will be displayed.

```
SETUP Passwords
                                              1/10
         USERNAME      PWD LEVE        TIME(min)
       1 JACK           -  INSTALL          15
       2 MARY           -  SETUP            15
       3                 *  ************     0
       4                 *  ************     0
       5                 *  ************     0
       6                 *  ************     0
       7                 *  ************     0
       8                 *  ************     0
       9                 *  ************     0
      10                 *  ************     0

   [ TYPE ]   LOGIN    LOGOUT             HELP    >
```

3 Move the cursor to your user name (Example: MARY).

4 Press F2, LOGIN to log in. The screen below will be displayed.

```
                                                   1/1



    --Set password for MARY                --

      Enter password: '                 '

                                        Alpha input  1
                                         Upper Case
                                         Lower Case
                                         Punctuation
   Old Value:                            Options
    ABCDEF   GHIJKL   MNOPQR   STUVWX   YZ_@*
```

5 Input your password and press the [ENTER] key. The screen below will be displayed.

```
SETUP Passwords
                                                   1/2
        Current user:         MARY
        Current level:        SETUP
      1 Default user timeout:        15 min

      2 Timeout occurs in:           15 min
      3 Log events:              DISABLE
      4 Number of users:             10







  [ TYPE ]   LOGIN   LOGOUT   PASSWRD    HELP    >
```

6 To change the timeout value, move the cursor to your user timeout value and input a new timeout value. The user timeout value may be in the range of 0 to 10080 minutes (7 days).

> ⚠ **CAUTION**
> If the user logs in to the teach pendant or CRT unit by setting the user timeout value to 0, no timeout is performed.

Step-b (In case that someone logs in)

1 Display the password setting screen.
- If the full menu is used, the procedure is as follows:
  - a Press [MENU] key
  - b Select SETUP.
  - c Press F1, [TYPE].
  - d Select Passwords.
- If the quick menu is used, the procedure is as follows:
  - a Press the MENU key.
  - b Select SETUP PASSWORDS.

2 Press F2, LOGIN. The screen below will be displayed.
In this example, JACK has already logged in.

```
SETUP Passwords
                                    2/10
        USERNAME     PWD LEVE       TIME(min)
     1@JACK           -  INSTALL        15
     2 MARY           -  SETUP          15
     3                 *  ************    0
     4                 *  ************    0
     5                 *  ************    0
     6                 *  ************    0
     7                 *  ************    0
     8                 *  ************    0
     9                 *  ************    0
    10                 *  ************    0

  [ TYPE ]   LOGIN   LOGOUT          HELP   >
```

⚠ CAUTION
The @, which appears when the user logs in, is a symbol indicating a login user name.

3 Move the cursor to your user name (Example: MARY).
4 Press F2, LOGIN to log in. Only one user can log in at a time. The following prompt is displayed on only the teach pendant or CRT window.

```
SETUP Passwords
                                    2/10
        USERNAME     PWD LEVE       TIME(min)
     1@JACK           -  INSTALL        15
     2 MARY           -  SETUP          15
     3                 *  ************    0
     4                 *  ************    0
     5                 *  ************    0
     6                 *  ************    0
     7                 *  ************    0
     8                 *  ************    0
     9                 *  ************    0
    10                 *  ************    0
    User JACK logged in. Force logout?[NO]
                              YES      NO
```

5 To log out the user who is logged in, press F4, YES. Otherwise, press F5, NO.
If F4, YES is selected, the screen below will be displayed.

```

                                   1/1



    --Set password for MARY           --

      Enter password: '               '

                                  Alpha input  1
                                   Upper Case
                                   Lower Case
                                   Punctuation
   Old Value:                      Options
  ABCDEF  GHIJKL  MNOPQR  STUVWX  YZ_@*
```

6 Input your password and press the [ENTER] key. The screen below will be displayed.

```
SETUP Passwords
                                        1/2
        Current user:        MARY
        Current level:       SETUP
      1 Default user timeout:      15 min

      2 Timeout occurs in:         15 min
      3 Log events:          DISABLE
      4 Number of users:          10

 [ TYPE ]  LOGIN  LOGOUT  PASSWRD  HELP  >
```

7 To change the timeout value, move the cursor to your user timeout value and input a new timeout value. The user timeout value may be in the range of 0 to 10080 minutes (7 days).

> ⚠ **CAUTION**
> If the user logs in to the teach pendant or CRT unit by setting the user timeout value to 0, no timeout is performed.

Procedure 9-17 Logging out

Condition

- A password has been set. (See the section, "Password Operations by the Install User".)
- The user is currently logged in. (See Procedure 9-16 "Logging in".)

Step

1 Display the password setting screen.
- If the full menu is used, the procedure is as follows:
  - a Press [MENU] key.
  - b Select SETUP.
  - c Press F1, [TYPE].
  - d Select Passwords.
- If the quick menu is used, the procedure is as follows:
  - a Press the MENU key.
  - b Select SETUP PASSWORDS.

2 Press F3, LOGOUT to log out. When the user logs out, the system returns to the operator level.

Procedure 9-18 Changing a password

Condition

- A password has been set. (See the section, "Password Operations by the Install User".)
- The user is currently logged in. (See Procedure 9-16 "Logging in".)

Step

1 Display the password setting screen.
- If the full menu is used, the procedure is as follows:
  - a Press [MENU] key.
  - b Select SETUP.
  - c Press F1, [TYPE].

d Select Passwords.

- If the quick menu is used, the procedure is as follows:
  a Press [MENU] key.
  b Select SETUP PASSWORDS The screen below will be displayed.

```
SETUP Passwords
                                      1/4
        Current user:        AAAA
        Current level:       INSTALL
      1 Default user timeout:      15 min

      2 Timeout occurs in:          4 min
      3 Log events:            DISABLE
      4 Number of users:           10

   [ TYPE ]  LOGIN  LOGOUT  PASSWRD  HELP  >
```

2 Press F4, PASSWRD. The screen below will be displayed.

```
                                       1/3



    --Set password for AAAA                 --

        Old password: '              '
        New password: '           Alpha input  1
        Verification: '            Upper Case
                                   Lower Case
                                   Punctuation
   Old Value:                      Options
    ABCDEF  GHIJKL  MNOPQR  STUVWX  YZ_@*
```

3 Input the current password and press the [ENTER] key.
4 Input a new password and press the [ENTER] key.
5 Input the new password again to check that the password first input is correct, and then press the [ENTER] key.
6 Press F3, LOGOUT. Then, follow Procedure 9-16 "Logging in" to log in and set the new password.

9.11.5 Password Configuration File

9.11.5.1 Overview

By using an XML file describing a password configuration, it is possible to set the password function. About the method to create the XML files for the password function, refer to 9.11.6 “XML Syntax for Password Configuration Files”. Furthermore, for information on creating password configuration XML files, refer to the supplied sample files. If the teach pendant or Internet Explorer is connected to the robot, the website of the robot can be displayed. From the website, select the link to "Error/Diagnostic files (MD:)" and then select "PASSCFG.DG (Password Configuration)". The screen below will be displayed.

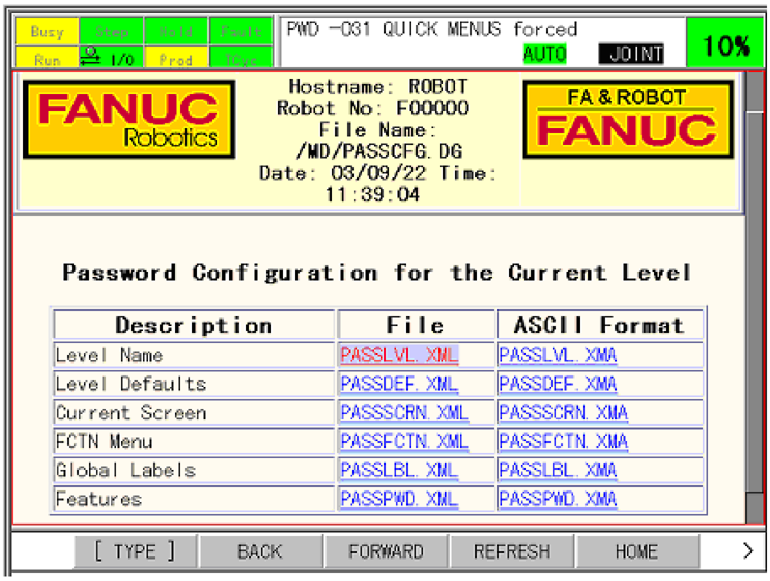


Fig. 9.11.5.1 Password configuration screen

Each of these links leads to an XML file sample. If the website of the robot cannot be displayed or if a file is to be copied, it is possible to copy a file from an MD:unit. To obtain a copy of a file, copy MD:PASSLVL.XML. A sample file describes the password configuration at the level of the user who is currently logged in.
A password configuration is saved as FRS:PASSWORD.DT. When the controller is turned on, this file is loaded so that passwords are configured. The FRS:PASSWORD.DT file can be saved and loaded as an application file on the file screen.
Table 9.11.5.1 "Password configuration settings" lists the settings that can be used for a password configuration. For information on creating a password configuration, see Procedure 9-19 "Password configuration".

Table 9.11.5.1 Password configuration settings

Setting	Explanation
VERIFY	Analyzes a selected XML file and writes alarms to a log file. Does not create actual configuration data. It is possible to select an XML file from an external storage unit. Set the external storage unit on the file screen. (See Chapter 8.)
IMPORT	Selects an XML file from an external storage unit. Set the external storage unit on the file screen. (See Chapter 8.) Analyzes the selected XML file to create configuration data.
EXPORT	Copies the current configuration data to an external storage unit, as password.xml. **NOTE** It is necessary to IMPORT the file first.

Procedure 9-19 Password configuration

Condition

- The user is logged in as an install user.

Step

1. Press [MENU] key
2. Select SETUP.
3. Press F1, [TYPE].
4. Select Passwords. The screen below will be displayed.

```
SETUP Passwords
                                             1/4
        Current user:        JOE
        Current level:       INSTALL
      1 Default user timeout:      15 min

      2 Timeout occurs in:          4 min
      3 Log events:            DISABLE
      4 Number of users:            10

[ TYPE ]  LOGIN  LOGOUT  PASSWRD  HELP  >
```

5 Press the [NEXT] key and then F6, CONFIG. The screen below will be displayed.

```
Configure Passwords

        Password configuration is performed
        by importing an XML file from the
        default device.

        The XML file can be verified first.

        The current configuration can be
        exported to password.xml

  DONE  VERIFY  IMPORT  EXPORT
```

9.11.6 XML Syntax for Password Configuration Files

9.11.6.1 Software parts ID, Screen ID

It is necessary to know software parts ID and screen ID in order to setup password for each screen. The following procedure is how to look into the software parts ID and screen ID of target screen.

Procedure 9-20 Looking into software parts ID and screen ID of target screen

Step

1 Make display to double

Press [SHIFT] key and display key. Display menu window like Fig. 9.11.6.1 (a) will be displayed. Select “2 Double” in display menu window. It is changed to be Fig. 9.11.6.1 (b).

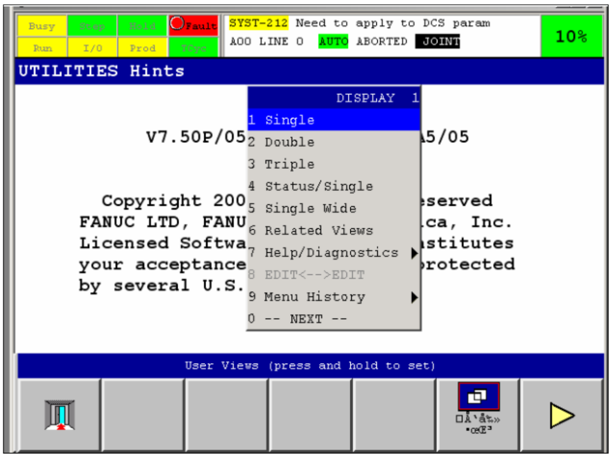


Fig. 9.11.6.1 (a)

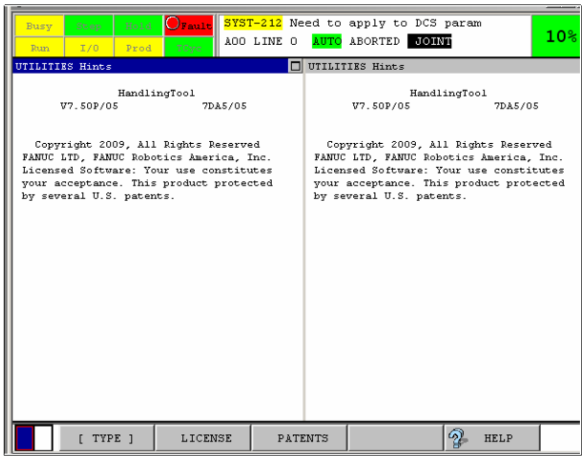


Fig. 9.11.6.1 (b)

2 Display system variable $TP_CURSCRN[1] in window 2

Active window can be switched by pressing display key. Open the second window. After active window is switched in the second window (right window), press menu key [0.NEXT], [6.SYSTEM], F1[TYPE], [SYSTEM Variable] to display the system variable screen in the second window. When $TP_CURSCRN is selected in system variable screen, $TP_CURSCRN[1 - 4] are displayed. Then, select $TP_CURSCRN[1]. The second window should be Fig. 9.11.6.1 (c). $SP_ID is software parts ID and $SCRN_ID is screen ID of the first window (left window).

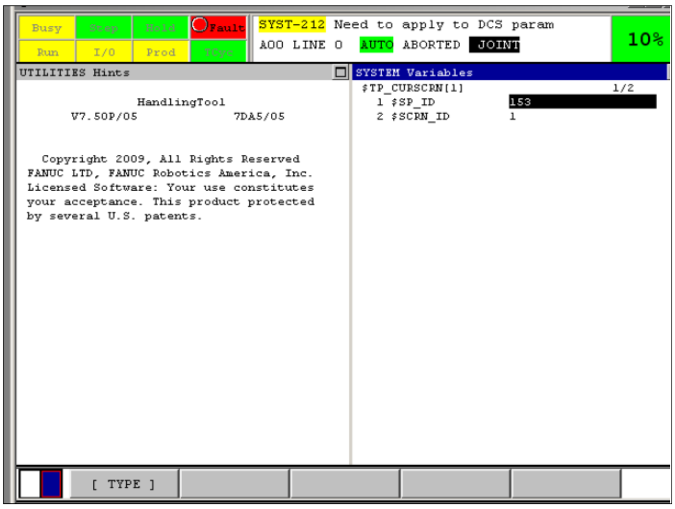


Fig. 9.11.6.1 (c)

3 Make window 1 to aim menu screen

Make active window to the first window and select your target screen in the first window. In this case, file screen is selected. In Fig. 9.11.6.1(d) , the first window is file menu screen, and you can see $SP_ID is 74 and $SCRN_ID is 1 in the second window. These ID's are for file menu screen's software parts.

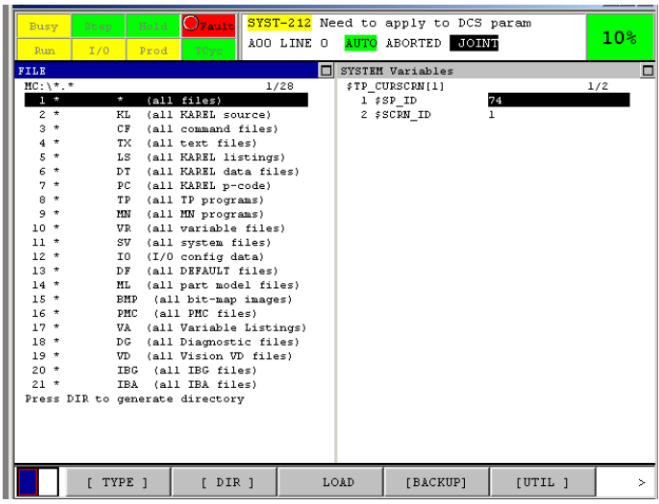


Fig. 9.11.6.1 (d)

9.11.6.2 Basic tag

XML file is composed only of the syntax which is called a tag. The beginning of tag is "<", and end of tag is ">". We explain about XML tag for password function below.

Basic tag must exist in case XML for password function in this controller.

- Starting command tag

```
<?xml version="1.0"?>
```

This tag should be at the top of your XML file.

- First tag

```
<PASSWORD>
```

This tag start XML for password function. Only one tag can be specified in your file.

- Last tag

```
</PASSWORD>
```

This tag indicate the end of XML for password function. Only one tag can be specified in your file.
So the beginning and end of all XML file should be like below.

```
<?xml version="1.0"?>
    <PASSWORD>
        .
        .
```

```
        .
        .
        .
    </PASSWORD>
```

9.11.6.3 Comment tag

This is comment. ALL comments are include within <!-- -->.

```
<!--Password configuration file-->
```

9.11.6.4 Level name tag

This tag defines the level name up to 12 characters. All levels can be renamed including the “INSTALL” level.

```
< LEVEL=”3” name=”Maintenance” lang=”eg”/>
```

- Write “LEVEL” in the beginning of tag and specify level parameter with “=”.The Default level names are as follows.

  Level 0— Operator
  Level 1— Program
  Level 2— Setup
  Level 3— Level 3
  Level 4— Level 4
  Level 5— Level 5
  Level 6— Level 6
  Level 7— Level 7
  Level 8— Install

- Each level tag defines one level name and one language. In order to define multiple levels or multiple languages, multiple tags are required.
- Specify level name to “name”. For example , “Level 3” is changed to “Maintenance”.
- Specify language to “lang”. But it should only be used in a multi-language system. If your controller is not multi-language system (menu select key, [6.SETUP], GENERAL, 2.Current Language is DEFAULT) this parameter is no need. Each language is defined blow. In example, it setup English and Japanese.

・eg-English	・gr-German
・jp-Kana	・sp-Spanish
・kn-Japanese	・ch-Chinese
・fr-French	・tw-Taiwanese

9.11.6.5 Local Labels tag

This tag defines the function key access locally within a screen for a given level. All level except INSTALL can be changed to allow/disallow function key access.
The following example is that F4, [BACKUP] in file menu screen is disallowed by using local label.

```
<LLABEL level="2" name="[BACKUP ]" lang="EG" sp_id="74" scrn_id="1" access="0"/>
```

- Write “LLABEL” in beginning of tag.
- Specify level parameter to “level”. The value of “level” and level name is below.

  Level 0— Operator
  Level 1— Program
  Level 2— Setup
  Level 3— Level 3
  Level 4— Level 4
  Level 5— Level 5
  Level 6— Level 6
  Level 7— Level 7
  Level 8— Install

- Specify function key’s name to “name”. Function key’s name is different form each language, so you need to specify name for each language. In an example, F4, [BACKUP] is disallowed. If you change the name of function key, it is possible to disallow other function key.
- Specify language to “lang”.
- Specify software parts ID and screen ID of the target screen you want to allow/disallow. The way how to look into software parts ID and screen ID of the target screen is described in 9.10.6.1. “Software parts ID Screen ID”.
- Specify 0 or 1 to ”access”.
  access=0 Function key or pull up entry cannot be pressed.
  access=1 Function key or pull up entry can be pressed.

9.11.6.6 Screen Access tag

This tag defines the screen access for a given level. All levels except INSTALL can be changed to allow/disallow write access.

```
< SCREEN level="2" sp_id="23" scrn_id="1" access="1" rw_access="0"/>
```

- Write “SCREEN” in beginning of tag.
- Specify level parameter to “level”.
- Specify software parts ID and screen ID of the target screen that you want to allow/disallow. The way how to look into software parts ID and screen ID of the target screen is descried in 9.10.6.1. “Software parts ID Screen ID”.
- Specify 0 or 1 to ”access”.
  access=0 Screen cannot be entered.
  access=1 Screen can be entered.
- Specify 0 or 1 to ”rw_access”.
  rw_access=0 If screen can be entered, it will only operate in read-only mode.
  w_access =1 If screen can be entered, it will only operate in read-write mode.

All levels except INSTALL can be changed to allow/disallow screen access or to allow/disallow write access. The exception is SETUP Passwords menu. A screen tag for this menu is ignored.

The default screen access is pre-defined by the system. Typically OPERATOR and LEVEL3-LEVEL7 have read-only access to all menus. PROGRAM level has read-write access to menus that program the robot. SETUP level has read-write access to menus that program and setup the robot. The “level” parameter defines the level.

If a screen is disallowed to display, the item of that screen in the menu displayed by MENU key or F1, [TYPE] key is displayed in gray and cannot be selected.

9.11.6.7 Features tag

This tag defines the access for a feature for a given level. All levels except INSTALL can be changed to allow/disallow access to a feature.

```
< PWD level=”2” const=”20” access=”0”/>
```

- Write “PWD” in the beginning of tag.
- Specify level parameter to “level”.
- Specify feature to “const”. The value of “const” and feature is below.

Table 9.11.6.7 The value of const and the feature

Value of const	Restriction	Description
const=1	KCL	Disallow KAREL command
const=2	I/O configuration	Disallow I/O configuration
const=3 to 17	Limitation about communication	
const=18	Display Menu	Disallow select menu display
const=19	SHIFT-COORD Menu	Disallow SHIFT-COORD Menu
const=20	Jog Access	Disallow jog access. Prohibits jogging w.p.r but x,y,z is allowed. JGFRM and Group 1 are forced
const=21	Production Speed Access	Prohibits lowering production speed below 80%

- Specify 0 or 1 to ”access”.
  access=0 Access to the feature is not allowed.
  access=1 Access to the feature is allowed.

9.11.6.8 Default menu type tag

This tag defines the default menu type for a given level.

```
<DEFMENU level=”3” type=”2”/>
```

- Write “DEFMENU” in beginning of tag.
- Specify level parameter to “level”.
- The type parameter defines the menu type.
  type=0 :Full menus
  type=1 :Quick menus
  type=2 :HMI Full menus
  type=3: HMI Quick menus

The default menu type for all levels in Full menus. All levels except INSTALL can be changed to have a different menu type. When a level changes such as when a user logs in, logs out, or times out, then the menu type will automatically change to the default menu type. By using this tag, it is possible to restrict the screen which can be selected depending on the password level.

9.11.6.9 Default screen tag

This tag defines the default screen for a given level. The default screen for all levels are not defined.

```
< DEFSCREEN level=”0” sp_id=”935” scrn_id=”1”/>
```

- Write “DEFSCREEN” in the beginning of tag.
- Specify level parameter to “level”.
- Specify software parts ID and screen ID of target screen that you want to allow/disallow. The way how to look into software parts ID and screen ID of the target screen is described in 9.11.6.1 “Software parts ID Screen ID”.

All levels except INSTALL can be changed to have a default screen. When a level is changed by user logs in, logs out, or time out, then the screen will automatically change to default screen. In case that the default screen is not defined, if the previous screen is allowed to display, the previous screen is displayed, if not, the first screen is displayed.

For example, in case that the software parts ID is set to 935 and the screen ID is set to 1, the default screen is set to the password screen. Following table describes the software parts ID and Screen ID for typical screen.

Table 9.11.6.9 Software parts ID and screen ID to set default screen

Default screen	Software parts ID	Screen ID
Not defined	0	0
SETUP Passwords	935	1
ALARM	18	1
SELECT	71	1
EDIT	64	1

9.11.6.10 FCTN/display menu access tag

The screen tag can also be used to define the access for each FCTN or Display Menu entry for a given level. All levels except INSTALL can be changed to allow/disallow.

```
< SCREEN level="3" sp_id="64" scrn_id="20" access="0"/>
```

- Write “SCREEN” in the beginning of tag.
- Specify level parameter to “level”.
- If you want to disallow FCTN menu, you need to specify software parts ID and screen ID of the FCTN menu. But, the way of 9.11.6.1 “Software parts ID, Screen ID” cannot used for FCTN. Software parts ID and screen ID of FCTN menu is below.

Table 9.11.6.10 (a) Software parts ID and screen ID for FCTN menu

FCTN menu	Software parts ID	Screen ID
ABORT(ALL)	20	1
Disable FWD/BWD	20	9
RELEASE WAIT	931	1
QUICK/FULL MENUS	64	20
SAVE	67	1
PRINT SCREEN	171	1
UNSIM ALL I/O	20	2
ENABLE/DISABLE HMI MENUS	1304	2
Diagnostic log	20	12
REFRESH PANE	381	21

- If you want to set Display Menu , Specify software parts ID and screen ID that you want to set Display Menu. But The way 9.11.6.1 “Software parts ID, Screen ID” cannot look into. Software parts ID and screen ID of Display Menu is below.

Table 9.11.6.10 (b) Software parts ID and screen ID for Display Menu

Display Menu	Software parts ID	Screen ID
Single	37	101
Double	37	102
Triple	37	103
Status/Single	37	104
EDIT↔EDIT	37	112
Menu History	37	109
User Views	37	107
Menu Favorite	37	108

- Specify 0 or 1 to ”access”.
  access=0 Access to the feature is not allowed.
  access=1 Access to the feature is allowed.

9.11.6.11 EDCMD access tag

The screen tag can also be used to define the access for each EDCMD entry in the teach pendant Editor.

```
< SCREEN level="3" sp_id="64" scrn_id="30" access="0"/>
```

- Write “SCREEN” in beginning of tag.
- Specify level parameter to “level”.
- If you want to disallow EDCMD, you need to specify software parts ID and screen ID of EDCMD. But, the way of 9.11.6.1 “Software parts ID, Screen ID” cannot used for EDCMD. Software parts ID and screen ID of EDCMD is below.

Table 9.11.6.11 Software parts ID and screen ID for EDCMD menu

EDCMD	Software parts ID	Screen ID
Insert	64	3
Delete	64	4
Find	64	5
Replace	64	6
Renumber	64	7
Copy	64	9
Comment	64	16
Undo	64	17

9.11.7 Password Log

If the install user enables the "Password log" item on the password setting screen, the password log displays the events below.

- Password event
- Programming event
- File operation event
- SpotTool application event

When user logins, program creations, and other events occur, they are all recorded in the password log.
The password log screen contains the item listed in Table 9.11.7 "Item on the password log screen".

Table 9.11.7 Item on the password log screen

Item	Explanation
Password Log	Monitors password logins and logouts. If password log is enabled, it is possible to check who is logged in and what changes have been made.

Only the install user can enable the "Log events" setting. For information on enabling the password log, see Procedure 9-21 "Enabling the password log".
All users can display the password log. For information on displaying the password log, see Procedure 9-22 "Displaying the password log".

Procedure 9-21 Enabling the password log

Condition

- The user is logged in as an install user. (Procedure 9-12 "Assignment a user name and a standard password for each password level")

Step

1 Press [MENU] key
2 Select SETUP.
3 Press F1, [TYPE].
4 Select Passwords. The screen below will be displayed.

```
SETUP Passwords
                                          1/4
        Current user:        AAAA
        Current level:       INSTALL
      1 Default user timeout:      15 min

      2 Timeout occurs in:          4 min
      3 Log events:           DISABLE
      4 Number of users:           10

  [ TYPE ]  LOGIN   LOGOUT  PASSWRD   HELP   >
```

5 Use the steps below to enable or disable the password log.
   a. Move the cursor to "Log events".
   b. To enable Log events, press F4, ENABLE.
   c. To disable Log events, press F5, DISABLE.

Procedure 9-22 Displaying the password log

Condition

- The install user has enabled Log events. (Procedure 9-21 "Enabling the password log")
- The user is logged in at any of the install, program, and setup levels.
- A full menu will be displayed.

Step

1 Press [MENU] key
2 Select ALARM.
3 Press F1, [TYPE].
4 Select Password Log. The screen below will be displayed.

```
Password Log
                                         1/7
    1 PWD -035 Login (BOB) INSTALL from CRT
    2 PWD -031 QUICK MENUS forced
    3 PWD -039 Logot (BOB) INSTALL from CRT
    4 PWD -052 Pwd Timeout (MARY) from Teac
    5 PWD -034 Login (MARY) SETUP from Teac
    6 PWD -031 QUICK MENUS forced
    7 PWD -038 Logout (MARY) SETUP from Tea

  [ TYPE ] [ VIEW ]         CLEAR   DETAIL
```

5 To display the details of a specific event, move the cursor to that event and press F5, DETAIL. The screen below will be displayed .

```
Password Log
    DETAIL Alarm
    PWD -035 Login (BOB) INSTALL from CRT/Ke

    WARN       16-SEP-10 19:32
    Password Log
    1 PWD -035 Login (BOB) INSTALL from CRT
    2 PWD -031 QUICK MENUS forced
    3 PWD -039 Logot (BOB) INSTALL from CRT
    4 PWD -052 Pwd Timeout (MARY) from Teac
    5 PWD -034 Login (MARY) SETUP from Teac
    6 PWD -031 QUICK MENUS forced
    7 PWD -038 Logout (MARY) SETUP from Tea

  [ TYPE ] [ VIEW ]         CLEAR   DETAIL
```

9.11.8 Screen Restrictions According to Password Level

You can confirm screen restriction according to password level by execution following procedure.

Procedure 9-23 Confirmation of screen restrictions according to password level

Step

1 Press [DISP] key with [SHIFT] key so that window menu will be displayed.
2 Select “8.Help/Diagnosis” and select “6.Password help” so that Password help screen is displayed. (Fig 9.11.8).

Busy Step Hold Fault Run I/O Prod TCyc T1 JOINT 10%

ROBOT (robot) Homepage

Read-Write Access
(LEVEL7, LEVEL6, LEVEL5, LEVEL4, LEVEL3, SETUP, PROGRAM, OPERATOR)

Top Level Menus	Sub Level Menus	Read-Write Access	Soft Part ID	Scrn ID
UTILITIES				
	Hints	(1,1,1,1,1,1,1,1)	153	1
	iRCalibration	(0,0,0,0,0,0,0,0)	1463	1
	Prog Adjust	(0,0,0,0,0,1,0,0)	930	1
	Program Shift	(0,0,0,0,0,1,0,0)	151	1
	Mirror Image Shif	(0,0,0,0,0,0,0,0)	150	1
	Tool Offset	(0,0,0,0,0,1,0,0)	155	1
	Frame Offset	(0,0,0,0,0,1,0,0)	191	1
	Angle entry shift	(0,0,0,0,0,0,0,0)	254	1
	Group Exchg	(0,0,0,0,0,0,0,0)	1200	1

BACK FORWARD REFRESH >

Fig. 9.11.8 Password help screen

You can confirm screen restriction according to password level as following.

- Top level Menus: This item means screen name on large category that is displayed when MENU key is pressed.
- Sub level Menus: This item means screen name on small category that is displayed when F1[TYPE] is pressed.
- Read-Write Access: This item means whether each password levels are permitted to change data on each sub level menus. Value of this item means following meaning for level7, level6, level5, level4, level3, SETUP level, PROGRAM level, OPERATOR level starting from left(INSTALL level is always permitted).
  1: This level user is permitted to change data. This level user isn’t limited operation.
  0: This level user isn’t permitted to change data. This level user is limited operation to change data and a part of function key.

EX) Read-Write Access of Program Adjust screen is (0,0,0,0,0,1,0,0) on Fig 9.11.8. In this case, only SETUP level user is permitted to change data on Program Adjust screen (INSTALL level is always permitted).

9.11.9 Password Auto Login Function

If the user name for the install level of the password function is one of the user names in the table below, a login is automatically performed at the install level if the corresponding condition is met. If the corresponding condition is no longer met, a logout is automatically performed and the system is set to the operator level.

User name	Condition
_AUTOLOGIN_0	The teach pendant enable switch is ON, and the 3-mode switch is in T1 or T2.
_AUTOLOGIN_1	The teach pendant enable switch is ON.
_AUTOLOGIN_2	The 3-mode switch is in T1 or T2.

The time at which a login or logout is automatically performed is when the system changes from a status in which the corresponding condition is not met to a status in which it is met or when the system changes from a status in which the corresponding condition is met to a status in which is not met.
Even if one of the user names above is set, it is possible to perform a login or logout on the password screen. In this case, an automatic login or logout may be performed later, when the corresponding condition is met or when it is not met.

If auto logins are enabled, the alarm message displayed if operation restrictions are imposed due to a password is not the ordinary one, "PWD-069 Operation password protected", but one of the alarm messages below.

User name	TP enable switch	3-mode switch	Alarm message
_AUTOLOGIN_0	ON	T1/T2	No restrictions.
		AUTO	PWD-72 Protected in AUTO mode
	OFF	T1/T2	PWD-71 Protected in TP disable
		AUTO	PWD-72 Protected in AUTO mode
_AUTOLOGIN_1	ON	Not applicable	No restrictions.
	OFF	Not applicable	PWD-71 Protected in TP disable
_AUTOLOGIN_2	Not applicable	T1/T2	No restrictions.
		AUTO	PWD-72 Protected in AUTO mode

If the password log is enabled, auto logins and auto logouts are recorded in the log, as follows:

PWD-073 'Auto login '"(%s) %s"
PWD-074 'Auto logout '"(%s) %s"

9.11.10 USB Password Function

By using USB password function, it is possible to login automatic only to insert USB device.

Procedure 9-24 Usage of USB Password Function

1 Press [MENU] key → 0.NEXT → 6.SYSTEM →
Set VARIABLES $PASSWORD.$ENB_PCMPWD to TRUE.

2 Select 6, SETUP.

3 Press F1, [TYPE] and select Passwords. The next screen will be displayed.

```
SETUP Passwords

         Password are inactive.
         Press LOGIN to activate.




   [ TYPE ]  LOGIN
```

4 Press F2, Login. The next screen will be displayed.

```
SETUP Passwords
                                          1/1
        USERNAME     PWD LEVEL       TIME(min)
      1               *  INSTALL            0




    Press ENTER to add INSTALL user.
   [ TYPE ]  LOGIN   LOGOUT          HELP    >
```

5 Setup user name and password at INSTALL level. If you need , setup other user's name and password. The screen will be displayed.

```
SETUP Passwords
                                         1/10
         USERNAME    PWD LEVE        TIME(min)
      1@EXEPERT       -  INSTALL           15
      2 USER1         -  PROGRAM           15
      3 USER2         -  PROGRAM           15
      4                 *  ************     0
      5                 *  ************     0
      6                 *  ************     0
      7                 *  ************     0
      8                 *  ************     0
      9                 *  ************     0
     10                 *  ************     0

   [ TYPE ]  LOGIN   LOGOUT          HELP    >
```

6 After login INSTALL level user . Move the cursor on the USER name that you want to use USB password function.

```
SETUP Passwords
                                          1/10
         USERNAME    PWD LEVE         TIME(min)
      1@EXEPERT       -  INSTALL            15
      2 USER1         -  PROGRAM            15
      3 USER2         -  PROGRAM            15
      4                 *  ************      0
      5                 *  ************      0
      6                 *  ************      0
      7                 *  ************      0
      8                 *  ************      0
      9                 *  ************      0
     10                 *  ************      0

   [ TYPE ]  LOGIN  LOGOUT         HELP    >
```

7 Press NEXT function key to switch into the next screen. (If $PASSWORD.$ENB_PCMPWD is not TRUE or don't login INSTALL level , USB is not displayed.)

```
              CLEAR  CLR_ALL   USB    HELP    >
```

8 Press F4, USB in situation without inserting USB device to USB slot . The next screen will be displayed.

```
       Insert USB device for password device.

                                 OK   CANCEL
```

9 Press F4, OK after insert USB you want to let password device into USB slot. The next screen will be displayed .

```
       USB device will be cleared, Proceed?

                                 OK   CANCEL
```

10 Press F4, OK if you have no problem.

> ⚠ **CAUTION**
> When USB device is made as password device, the contents of USB device is lost. Check the contents of USB device in advance.

11 The next screen will be displayed. USB device inserted to USB slot is recognized as password device of USER name that the cursor is stayed in procedure 6.

```
       Password device is created successfully.
```

12 If next conditions are met, it is possible to login automatic only to insert USB device recognized as password device.
   1 $PASSWORD.$ENB_PCMPWD is TRUE.
   2 The USER name corresponding to the password device is setup in the controller.
   3 All users are logout.

13 If USB device recognized as password device is pull or timeout time is over, user is logout automatically.

9.11.11 GI Password Login function

It is possible to login to a password user according to the value of setup group IO(GI). Devices(such as teach pendant, CRT/key board) login by the way except for this function won't login by this function.
And, it is possible to setup a Digital IO(DO) as Watch Dog signal too.
This function comes true by running the following KAREL(GIPLOGIN), which was prepared in advance, in background, and the following KAREL will be loaded by repower after $PASSWORD.$ENB_LOGIN becoming TRUE.

- GIPLOGIN: Control Password users Login/Logout by the value of GI.
- RUNGILOG: Run the KAREL GIPLOGIN in background. So it is necessary to make this KAREL run during startup by registering RUNGILOG to "COLD START Autoexec program"/"HOT START Autoexec program" etc.
- GIPSETUP: KAREL for setup GI Password Login function. It is possible to setup on KAREL Var screen after selecting this KAREL.

This function can be used by version 28 of 7DF1 series or later, version 07 of 7DF3 series or later.

Procedure 9-25 Usage of GI Password Login Function

Condition

- Password users have been setup.
- Group IO(GI) used for GI Password Login Function has been assigned.
- In case using Watch Dog signal, Digital IO(DO) used for Watch Dog signal has been assigned.

Step

1 Press [MENU] key→0.Next→6.Syestem→Variables and set $KAREL_ENB 1 and set $PASSWORD.$ENB_LOGIN TRUE.
2 Please re-power.
3 Select the KAREL program GISETUP on Select screen.
4 Display KAREL Vars screen. The following screen will display.

```
DATA KAREL Vars
 GIPSETUP                              1/ 4
  1 U_GIPORT_NUM          1
  2 U_WDOGP_NUM           0
  3 U_GI_VALUE          [200] of INTEGER
  4 U_WDOG_TIME           500

   [ TYPE ]                                >
```

5 On the above KAREL Vars screen, setup Group IO(GI) used as GI Password Login by setup the KAREL variables U_GIPORT_NUM、U_GI_VALUE[1-200] according to the following descriptions. (Re-power needs to apply the setting)

- U_GIPORT_NUM: Logic port number of GI used for GI Password Login(But 0 or uninitialized means this function is disabled).
- U_GI_VALUE[1-200]: This value of GI for each login Password user. Array number of U_GI_VALUE means Password user number. However GI's value is 0 means Logout.

```
DATA KAREL Vars
 GIPSETUP                                1/ 4
  1 U_GIPORT_NUM            1
  2 U_WDOGP_NUM             0
  3 U_GI_VALUE            [200] of INTEGER
  4 U_WDOG_TIME             500

    [ TYPE ]                                >
```

```
DATA KAREL Vars
 U_GI_VALUE                                1/ 200
  1 [1]          1
  2 [2]          2
  3 [3]          3
  4 [4]          4
  5 [5]          0
  6 [6]          0
  7 [7]          0
  8 [8]          0

    [ TYPE ]                                >
```

For example, when setup as the above screen and GI[1] becomes 3、 a Password user, which was setup as password user number 3 on password screen, will login, and when GI[1] becomes 0, the password user will logout.

6 When you want to setup Watch Dog signal, on the above KAREL Vars screen, setup Digital IO(DO) used for Watch Dog signal of GI Password Login function by setup the KAREL variables U_WDOGP_NUM:, U_WDOG_TIME according to the following descriptions. (Re-power needs to apply the setting)

- U_WDOGP_NUM:DO's logical port number used for Watch Dog signal of GI Password Login function. (But 0 or uninitialized means Watch Dog signal is disabled).
- U_WDOG_TIME :Timeout time of Watch Dog signal(Unit is ms). Minimum is 500ms and Timeout time of Watch Dog signal will be 500ms even if you set a value less than 500..

```
DATA KAREK Vars
 GIPSETUP                                1/ 4
  1 U_GIPORT_NUM            1
  2 U_WDOGP_NUM             2
  3 U_GI_VALUE            [200] of INTEGER
  4 U_WDOG_TIME             500

    [ TYPE ]                                >
```

For example, when setup as the above screen, DO[2] becomes Watch Dog signal of GI Password Login function, and when DO[2] is not changed for 500ms, Watch Dog signal is regarded as no response, and Password user, which login by GI Password Login function, will logout.

7 Setup RUNGILOG to "HOT START Autoexec program" when HOT START is enabled, to "COLD START Autoexec program" when HOT START is disabled.

8 Please re-power.

9.12 BACKGROUND LOGIC

Overview

By using a background operation, it is possible to execute a TP program containing available instructions in the background. The program is executed repeatedly from the beginning to the end. This execution is not influenced by emergency stops, holds, or alarms.

Available instructions for background logic are written in “9.11.2 Available instructions for Background Logic”.
To execute background logic, use the procedure described in Procedure 9-26 "Setting up a background operation".

9.12.1 Execution Modes and Scan Time

A background operation can be in two execution modes, standard mode and high-level mode.

- In standard mode, all mixed logic commands can be executed in the background. In each ITP, 600 item are scanned. (One ITP is usually eight milliseconds.) "Item" as used here refer to data items and operators. If there are more than 600 items, scanning time increases. For example, if there are 1600 operations, processing time is 24 milliseconds. If programs of high-level mode are executed, number of items can be scanned in each ITP is (600 – number of high-level mode item).
- In high-level mode, all mixed logic commands can be executed in the background. Programs of high-level mode can be executed certainly in every ITP. (One ITP is usually eight milliseconds.) In each ITP, 540 items can be scanned. Please separate some statements from standard mode program and execute by high-level mode, if these statements must be execute at high frequency.

Table 9.12.1 Execution modes for a background logic

Mode	Maximum allowable number of items	Scanning time
Standard mode	No restriction	(Number of items in standard mode / (600 – Number of items in high-level mode)) * scanning time of high-level mode). The number of items means the total number of items in the mode of background logic programs.
High-level mode	540	Eight milliseconds

- Up to eight programs can be executed at the same time as a background operation.
- If number of high-level mode item exceeds maximum number, alarm “INTP-651 High level mode programs too big” occurs and the background program stops. In this case, please decrease number of high-level mode item.

NOTE
If number of items in high-level mode is increased, standard mode requests longer scan time.

- While a program is being executed in the background, the program cannot be edited or executed as an ordinary task. If the program is not executed in the background, the program can be executed as an ordinary task.
- If a program is being executed in the background, a program with the same name cannot be loaded in overwrite mode.

- If the controller power is turned off when a program is executed as a background operation, the program is automatically executed in the same execution mode at the next power up.
- The execution of a background operation takes one millisecond per ITP. The execution of a background operation may influence the cycle time of an ordinary program.
- To reduce the execution time of a background operation, reduce the value of $MIX_LOGIC.$ITEM_COUNT. $MIX_LOGIC.$ITEM_COUNT defines the number of items to be processed per ITP. The standard value is 600. This value should not be set over 600 because this value may influence the cycle time of an ordinary program.

9.12.2 Available Instructions for Background Logic

For Background operation, instructions in Table 9.12.2 are available.

Table 9.12.2 Available instructions for a background logic

Available instructions	Assignment statements Conditional expression JMP LBL[ ] LBL[ ] Run Select UALM[ ]
Available data	F[ ], M[ ], DI[ ], DO[ ], AI[ ], AO[ ], GI[ ], GO[ ], SI[ ], SO[ ], UI[ ], UO[ ], RI[ ], RO[ ] WI[ ], WO[ ] ON, OFF R[ ], PR[i.j], AR[ ] Constant Parameter Timer, timer overflow
Available operators	(,), !, AND, OR, =, <>, <, <=, >, >=, +, —, *, /, DIV, MOD

- M[ ] cannot be specified on the left side of an assignment statement in a background operation.
- If the program contains other than assignment statements, "INTP-443 Invalid item for Mixed Logic" is generated at execution time.
- In a background operation, assignment statements with conditional expressions can be executed. If the result of a conditional expression is OFF, the corresponding assignment statement is not executed. In the example below,
  IF (DI[1]), DO[1]=(DI[2])
  If DI[1] is ON, DI[2] is assigned to DO[1]; and if DI[1] is OFF, DO[1] is not changed.
- Brackets have to be used for conditional expression. Condition expression without brackets cannot be executed by background logic.
  Example: Following logic cannot be executed.
  IF DI[1] = ON, JMP LBL[1]

Please change as following.
IF (DI[1] = ON), JMP_LBL[1]

- In a background operation, pulse commands can be used. By combining a pulse command with a conditional expression, it is possible to create a off delay timer. An example is given below.
  IF (DI[1]), DO[1]=Pulse 1sec
  If DI[1] remains ON for longer than one second, DO[1] is ON for one second. If DI[1] becomes OFF before one second elapses, DO[1] becomes OFF immediately. While DI[1] is OFF, this statement does not set DO[1].
  To keep DO[1] ON for one second even if D1[1] becomes OFF within one second, use the statement below.
  F[1]=(DI[1] OR (F[1] AND DO[1]))
  IF (F[1]), DO[1]=Pulse 1sec
- If no time is specified with a pulse command, one scan pulse is assumed in the execution of a background operation. An example is given below.
  IF (DI[1]), DO[1]=Pulse
  In this case, if DI[1] is changed from OFF to ON, DO[1] becomes ON for one scan. This can be used for edge detection.
- If executed in an ordinary way, a time-less pulse command uses the time set for $DEFPULSE, but if the command is executed in the background, a different time is used.
- Target LBL of JMP statement must be set at line after the JMP statement. If the target LBL is before JMP, the program cannot start and alarm "INTP-443 Invalid item for Mixed Logic" occurs.

NOTE

When programs included JMP statement is executed in background logic, scan time of background program may be changed by branch of program execution.

- If the program that is executed by RUN instruction in background logic is required to pause, the program is not paused but aborted.
- In the following case, the program cannot be executed by RUN instruction in background logic. The alarm "INTP-651 High level mode programs too big" occurs and the background logic is stopped.
  - The program executed by RUN instruction has the motion group.
  - The program executed by RUN instruction is executing.
  - Background logic tries to start same program by RUN instruction in consecutive scans repeatedly.
    Please prevent to execute RUN instruction consecutively by combining IF instruction.

  Example : In case that the program SUB_A is executed at rising edge of DI[1].

```
Background Program1                      SUB_A
…………………                                   …………………
F[2] = ( DI[1] AND !F[1] )                …………………
F[1] = DI[1]                              F[3]=OFF
IF ( !F[2] OR F[3] ) JMP LBL[1]          [End]
   F[3] = ON
   RUN SUB_A
LBL[1]
…………………
[End]
```

- If UALM[] instruction is executed in background logic, the program in background logic is not stopped.
- If a UALM[] instruction is executed in multiple consecutive scans, the user alarm is issued at the first scan. After that, the UALM[] instruction is ignored. In this case, if the alarm number in UALM instruction is specified indirectly, when the alarm number is changed, the user alarm is issued.
- The user alarm can be issued only once per one scan (1 ITP). If the multiple UALM[] instructions are executed in one scan, only the first user alarm is issued in the scan. The other user alarm are issued after next scan. If multiple UALM[] instructions are executed in a short period of time, it may take a long time to output all alarm.

9.12.3 Execution and Setting of Background Logic

Using the Background Logic screen, it is possible to set up and execute a program as a background operation. For information on each of the items on the Background Logic screen, see Table 9.12.3 (a). Table 9.12.3 (b) lists the operations on the Background Logic screen.

Table 9.12.3 (a) Items on the background logic screen

Item	Explanation
PROGRAM	Enter the name of the program to be executed as a background operation.
STATUS	The statuses of the background operation programs will be displayed . • Stop: The program is stopped. • Running: The program is executed in standard mode. • Running(High): The program is executed in high-level mode.
MODE	This item is used to select an execution mode. • Normal: Always executes the program in standard mode. • High: Always executes the program in high-level mode.

Table 9.12.3 (b) Operations on the background logic screen

Function key	Explanation
RUN	Press this key to execute a program as a background operation. An error occurs if the program contains any statement that cannot be used in a background operation.
STOP	Press this key to stop the execution of a program in the background.

Procedure 9-26 Setting up a background operation

Step

1 Press [MENU] key
2 Select “6 SETUP”.
3 Press F1, [TYPE].
4 Select BG Logic. The screen below will be displayed.

```
Background logic
    Normal mode scan time:   8msec      1/8
         PROGRAM         STATUS          MODE
       1 LOGIC1          Running         Normal
       2 LOGIC2          STOP            Normal
       3 LOGIC3          Running (High)  High
       4                 Stop            Normal
       5                 Stop            Normal
       6                 Stop            Normal
       7                 Stop            Normal
       8                 Stop            Normal

   [ TYPE ]    RUN      STOP    [CHOICE]   CLEAR
```

5 Enter the name of the program to be executed as a background operation in the PROGRAM column. In the STATUS column, the statuses of background operation programs will be displayed.
- Stop : The program is stopped.
- Running : The program is executed in standard mode.
- Running(High) : The program is executed in high-level mode.

6 In the MODE column, select a mode.
- Normal : Executes the program in standard mode.
- High : Executes the program in high-level mode.

7 To execute the program as a background operation, press F2, RUN.

8 To stop the program in the background, press F3, STOP.

9.12.4 Other Instructions and Functions

Flags

A flag (F[ ]) is an internal I/O port that can be read and set. A flag is not connected to an actual I/O device, and is similar to a Boolean type variable.

The value of a flag can be recovered with the power failure recovery function (hot start) as the same as other output ports (DO, for example).

All flags are set to OFF in the events below.

- Cold start
- Controlled start
- INIT start
- Hot start with the I/O allocation being changed
- Hot start with the I/O configuration being changed

The flag can be assigned to DI, DO, UI, UO, GI, and GO as rack 34, slot 1, start points 1 to 1024.

> ⚠ **WARNING**
> If a flag is allocated to UI, the UI signals are controlled by changing the value of the flag from the teach pendant and a program, thereby making it possible to start the program and perform other operations. If allocating a flag to UI, design the system by satisfactorily considering safety. Otherwise, unit damage and personal injury may result.

To display the Flag screen, use the procedure below.

1 Press [MENU] key.
2 Select “5 I/O”.
3 Press F1, [TYPE].
4 Select Flag. The following screen will be displayed.

```
Flag
              #   STATUS                  1/1024
        F[    1]      ON  [                  ]
        F[    2]     OFF  [                  ]
        F[    3]     OFF  [                  ]
        F[    4]      ON  [                  ]
        F[    5]      ON  [                  ]
        F[    6]      ON  [                  ]
        F[    7]      ON  [                  ]
        F[    8]      ON  [                  ]
        F[    9]     OFF  [                  ]
        F[   10]     OFF  [                  ]
        F[   11]     OFF  [                  ]

   [ TYPE ]  DETAIL              ON       OFF
```

On this screen, the values of the flags can be changed.

5 To display the Port Detail screen, press F2, DETAIL. The following screen will be displayed.

```
Outputs
    Port Detail                        1/1

        Flag               [    1]



       1 Comment:  [                      ]




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

On this screen, the comment on a flag can be changed.

Markers

On the Marker screen, the statuses of markers can be monitored.

A marker (M[ ]) is similar to a flag, but the value of a marker cannot be edited directly. If a marker (M[ ]) is specified on the left side of an assignment expression (=) in a TP program, and the statement is executed, the expression is internally defined as a background operation and is executed repeatedly. The marker (M[ ]) always contains the result of the expression.

As standard, the marker function is disabled and the Marker menu does not appear, so that M[ ] cannot be taught in a TP program. To use the marker function, set $MIX_LOGIC.$USE_MKR to TRUE.

Example:

M[1]=(DI[1] AND DI[2])

After this line is executed in an ordinary TP program (not in a background operation), M[1] always contains the result of the expression on the right side. If both DI[1] and DI[2] are ON, M[1] is ON; otherwise, M[1] is OFF.

- If the marker assignment statement is executed in an ordinary TP program, the statement is registered as a background operation. The statement is executed as a background operation until another expression redefines the marker. Because the statement is a background operation, the execution of the statement is not stopped even if the program is paused or stopped.
- As standard, there are eight markers (M[1] to M[8]). The number of markers can be changed with the system variable $MIX_LOGIC.$NUM_MARKERS. The maximum allowable number of markers is 100. One marker occupies a permanent memory area of 300 bytes.
- The scan time in which to calculate marker assignment statements is the same as that in a background operation in standard mode. If marker assignment statements are defined, the scan time in a background operation is influenced.
- To stop calculation, clear the defined marker expression. To clear a defined expression, either execute an CLEAR operation on the Marker detail menu or execute the line below in a TP program.
  M[1]=()
- If a marker is not allocated in an expression, and the marker is used in another statement, "INTP-438 I/O Read I/O value failed" is generated at the time when the marker value is read.
- It is not possible to specify M[ ] on the left side of an assignment statement in a background operation.

To display the Marker screen, use the procedure below.

1 Press [MENU] key.
2 Select “5 I/O”.
3 Press F1, [TYPE].

4 Select Marker. The following screen will be displayed.

```
Marker
              #   STATUS                 1/8
        M[    1]      ON  [                  ]
        M[    2]     OFF  [                  ]
        M[    3]     OFF  [                  ]
        M[    4]      ON  [                  ]
        M[    5]      ON  [                  ]
        M[    6]      ON  [                  ]
        M[    7]      ON  [                  ]
        M[    8]      ON  [                  ]

   [ TYPE ]  DETAIL
```

On this screen, the values of the markers can be changed.

5 To display the Port Detail screen, press F2, DETAIL. The screen below appears.

```
Port detail
                                      1/6
     Marker                 [    1]

     Comment:  [                      ]

     Expression:
       M[1]=((DI[1] OR DI[2]) AND !
       DI[3])

     Monitor:
       M[1]                 ON
       DI[1]                OFF

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

On this screen, the comment on a marker can be changed.

- On the Port Detail screen, defined expressions appear. To clear a defined expression, press F5, CLEAR. When the message below appears,

```
      Clear expression?

                                 YES      NO
```

Press F4, YES.

- The current values of all the data items in the defined expressions are displayed in the Monitor area.

TC_ONLINE

TC_ONLINE is similar to a marker. The TC_ONLINE command defines an expression, which is, in turn, calculated as a background operation. While the result of the expression remains OFF, the execution of all TP programs and KAREL programs having motion groups is stopped.

As standard, the TC_ONLINE function is disabled, and the TC_ONLINE menu does not appear, so that TC_ONLINE commands cannot be taught in a TP program. To use the TC_ONLINE function, set $MIX_LOGIC.$USE_TCOL to TRUE.

The following shows an example.

TC_ONLINE (DI[1] AND DI[2])

While DI[1] or DI[2] remains OFF after this line is executed, all TP programs are stopped. See Table 9.12.4 .

Table 9.12.4 TC_ONLINE commands

TC_ONLINE (...)	Defines a specified mixed logic command as a TC_ONLINE expression and enables TC_ONLINE.
TC_ONLINE DISABLE*	Disables TC_ONLINE. If TC_ONLINE is disabled, the TP program is not stopped with TC_ONLINE (...).
TC_ONLINE ENABLE*	Enables TC_ONLINE. It is used to enable TC_ONLINE that has been disabled with TC_ONLINE DISABLE.

* As standard, TC_ONLINE DISABLE and TC_ONLINE ENABLE cannot be used. To use these commands, set $MIX_LOGOC.$USE_TCOLSIM to FALSE.

- When TC_ONLINE (...) is executed, the specified expression is defined as a TC_ONLINE expression. While TC_ONLINE is enabled, the defined expression is calculated as a background operation. While the result of the expression is OFF, all TP programs and KAREL programs having motion groups are stopped.
- If a program is started while the result of the TC_ONLINE expression is OFF, the program is immediately paused. Except Shift + BWD, starts of all types are influenced by TC_ONLINE.
- The only case in which a program can be executed when the result of the TC_ONLINE expression is OFF is when the program is executed backward using Shift + BWD.
- Even if the scan time in a background operation is longer than an ITP, TC_ONLINE expressions are calculated in each ITP. (One TIP is usually eight milliseconds.)
- A program that does not have a motion group and for which "ignore pause" is enabled is not paused even if the result of the TC_ONLINE expression is OFF.
- If $MIX_LOGIC.$USE_TCOLSIM is TRUE (standard), TC_ONLINE DISABLE and TC_ONLINE ENABLE cannot be taught on the Edit menu. It is necessary to change the TC_ONLINE ENABLE/DISABLE setting on the TC_ONLINE menu. In this case, when the execution of an operation statement is completed, TC_ONLINE is automatically enabled. This means that TC_ONLINE is disabled only in those brief cases in which the operator operates the robot.
- If $MIX_LOGIC.$USE_TCOLSIM is FALSE, TC_ONLINE DISABLE and TC_ONLINE ENABLE can be taught on the Edit menu. The TC_ONLINE ENABLE/DISABLE setting is not automatically changed even after the execution of an operation statement is completed.

To add a TC_ONLINE command, use Procedure 9-27. To display the TC_ONLINE menu, use Procedure 9-28.

Procedure 9-27 Adding a TC_ONLINE command

Step

1. On the Edit menu, press F1, [INST] to move to the Instruction menu.
2. Select TC_ONLINE.

   If $MIX_LOGIC.$USE_TCOLSIM is TRUE, select items and create statements in an ordinary way.
   If $MIX_LOGIC.$USE_TCOLSIM is FALSE, follow the steps:

   a Select “1(...)”.
   b Select items and create statements in an ordinary way.

3. To change a TC_ONLINE statement, press F4, [CHOICE] at the first "(" of the TC_ONLINE statement, and select “2 ENABLE”.

Procedure 9-28 Displaying a TC_ONLINE command

Step

1. Press [MENU] key
2. Select “5 I/O”.
3. Press F1, [TYPE].
4. Select TC_ONLINE. The screen below appears.

```
TC_ONLINE
                                          1/8
     Status:            ON
     Enable:            TRUE
     Expression:
       ((DI[1] OR DI[2]) AND !DI[3] AND
        !(DI[4] AND DI[5]))

     Monitor:
       DI[1]                ON
       DI[2]                OFF
       DI[3]                ON
       DI[4]                OFF

   [ TYPE ]                      TRUE    FALSE
```

The "Enable" line indicates whether TC_ONLINE is currently enabled. This item can be changed on this menu.
In the "Status" line, the status of TC_ONLINE is indicated. This is the result of a defined expression. The defined expression is indicated in the Expression space.

5. To clear the defined expression, move the cursor to “Monitor”.
   a. Press F5, CLEAR. The following prompt will be displayed.

```
      Clear expression?

                                  YES      NO
```

   b. Press F4, YES.

9.12.5 Backup for Background Logic

All data for background logic commands are saved as described below.

- Background logic programs are saved to TP files.
- Settings on the "Background Logic" menu are saved to MIXLOGIC.SV. In MIXLOGIC.SV, the values of the system variables are saved.
  - $MIX_LOGIC
  - $MIX_BG
  - $MIX_MKR
  - $DRYRUN
  - $DRYRUN_PORT
  - $DRYRUN_SUB
- Comments on flags and markers are saved to DIOCFGSV.IO.
- If DI/O, UI/O, or GI/O are allocated to flags , their allocations are saved to DIOCFGSV.IO.

9.13 ORIGINAL PATH RESUME

9.13.1 Original Path Resume

Overview

Original Path Resume is a feature that, after program pause (due to Hold or E-Stop, etc.), allows the robot to return to its stop position before resuming the program, even if the robot has moved since the program is paused. After return to the stop position, the robot continues the rest of the TP program, resulting in a path identical to the original one.
The move from the current position to the stop position on the original path is called the “Restart Move”, and the next move from the stop position is called the “Resume Move”.

The stop position (destination of Resume Move) is the last commanded position on the original path. That is, the stop positions of the following two cases are different from each other:

- Decelerating stop ... ex) Stop by HOLD input, or alarms which don’t cause servo off
- Immediate stop ... ex) Stop by alarms which cause servo off like E-Stop

For details of each case, refer to the subsections described below.

Setting of Original Path Resume

Original Path Resume is enabled using the following system variable:

$SCR.$ORG_PTH_RSM
TRUE: Original Path Resume is Enabled
FALSE: Original Path Resume is Disabled

This feature is enabled by default. But it may be disabled if some other specific option is loaded or enabled.

> ⚠ **CAUTION**
> Don’t enable this feature if it has been disabled without manual setting change.

Resume Path after Decelerating Stop (HOLD, etc.)

If a Decelerating Stop occurs (caused by HOLD input or Stop severity alarm, etc.), the robot will decelerate to stop along the path and pause the TP program. The user might then jog the robot to another position to correct the cause of the alarm. When the TP program is resumed:

- If Original Path Resume is DISABLED, the robot will dispose the original path and generate a new path from the current position. As the result, the new path may be different from the original one.
- If Original Path Resume is ENABLED, the robot will first execute a Restart Move to the Stop position, and then resume the TP program.

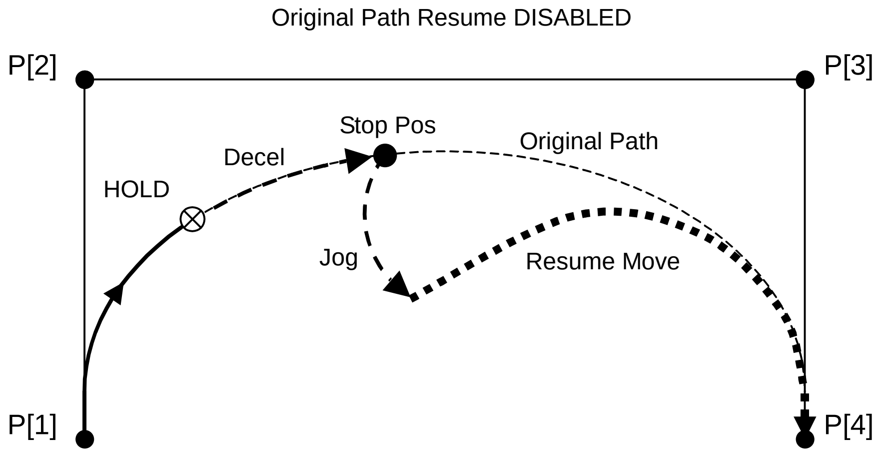


Fig. 9.13.1(a) Resume Path after Decelerating Stop with Original Path Resume DISABLED

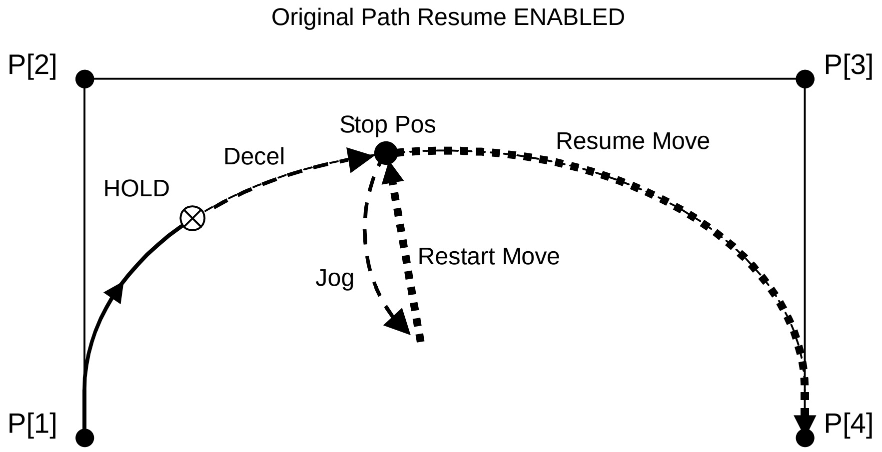


Fig. 9.13.1(b) Resume Path after Decelerating Stop with Original Path Resume ENABLED

Resume Path after Immediate Stop (E-Stop, etc.)

If an Immediate Stop occurs (caused by Emergency Stop or other high severity alarms etc.), servo power is removed and the robot stops. This possibly causes the robot to deviate from the normal path depending on the robot's payload mass, speed, and the direction of gravity. When the TP program is resumed:

- If Original Path Resume is DISABLED, the robot will dispose the original path and generate a new path from the current position. As the result, the new path may be different from the original one.
- If Original Path Resume is ENABLED, the robot will first execute a Restart Move to the position at which the alarm occurred, and then resume the TP program.

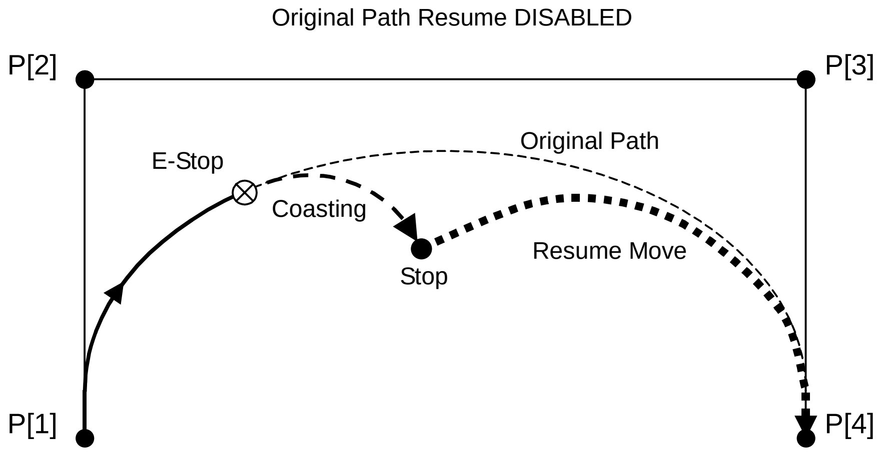


Fig. 9.13.1(c) Resume Path after Immediate Stop with Original Path Resume DISABLED

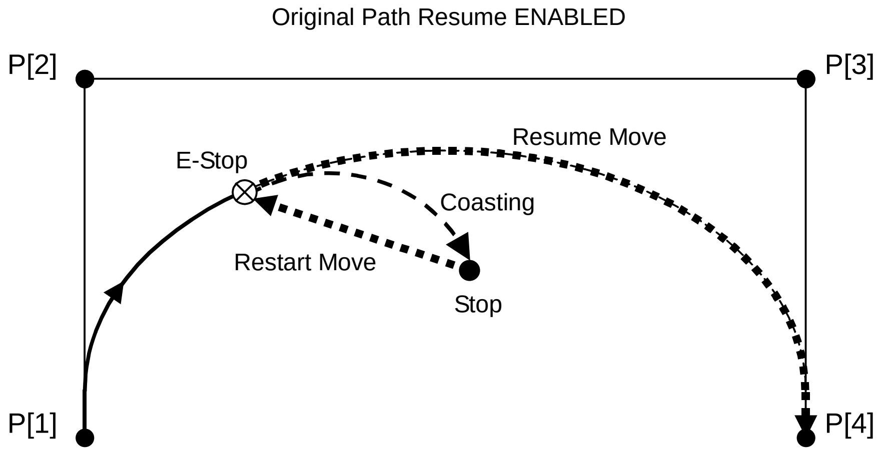


Fig. 9.13.1(d) Resume Path after Immediate Stop with Original Path Resume ENABLED

Resume Path in special cases

Pause and jog feed during Restart Move

Even if the last Restart Move was paused and jog feed was executed, the next Restart Move will directly go to the original Stop position, not go to the Stop position of the last Restart Move.

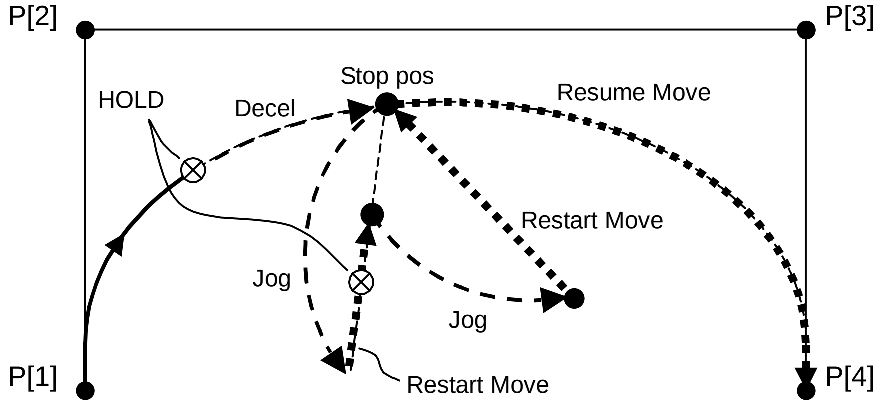


Fig. 9.13.1(e) Resume Path after pause and jog feed during Restart Move

Backward execution during pause

If robot moves by Backward (BWD) execution during a pause, the Stop pos for the Original Path Resume is disposed. When the TP program is resumed, regenerate new path from the current position. As the result, the new path may be different from the original one.

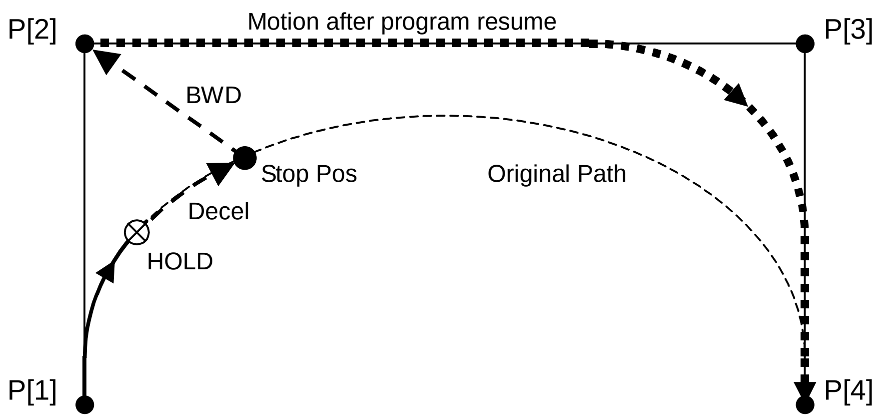


Fig. 9.13.1(f) Resume Path after BWD execution

Constant Path Disabled

If Constant Path is disabled ($CPCFG.$CP_ENABLE = FALSE), the path of Resume Move is generally not on the original path even if Original Path Resume is enabled.

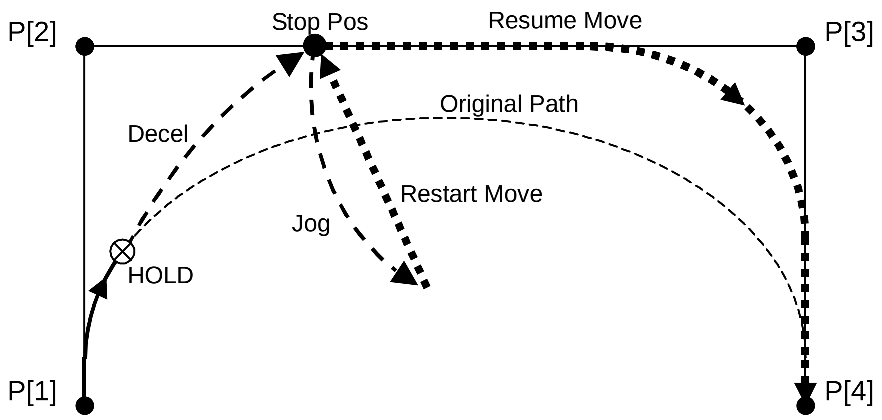


Fig. 9.13.1(g) Resume Path with Constant Path disabled

Limitations and usage notes of Original Path Resume

- It is possible for the robot to get to a position in which the Restart Move may cause alarms such as “Stroke Limit”, in which case operator intervention will be necessary to resume the program.
- Because the Restart move starts from the current position , it is possible for the robot to hit something if the path is not clear between the current position and the stop position. For example, the robot can be jogged to a position that will cause a collision with a fixture when the Restart Move executes. Therefore, make sure there is no obstacle along the Restart Move.
- If Continuous Turn option is enabled, Original Path Resume automatically becomes disabled ($SCR.$ORG_PTH_RSM = FALSE).
- In the following motion, the path of Resume Move is generally not on the original path even if Original Path Resume is enabled.
  - Arc welding motion

- Servo Gun motion with pressurization
- Search Motion of Touch Sensing
- Tracking program motion of Line Tracking
- Weaving motion

9.13.2 Resume Offset

Overview

Resume Offset if an extension feature of Original Path Resume. This feature allows the Restart move to go to a previous position of the path, "offset" from the normal stop position. Processes those are continuous along the path, such as cutting, welding, or dispensing, use this feature.

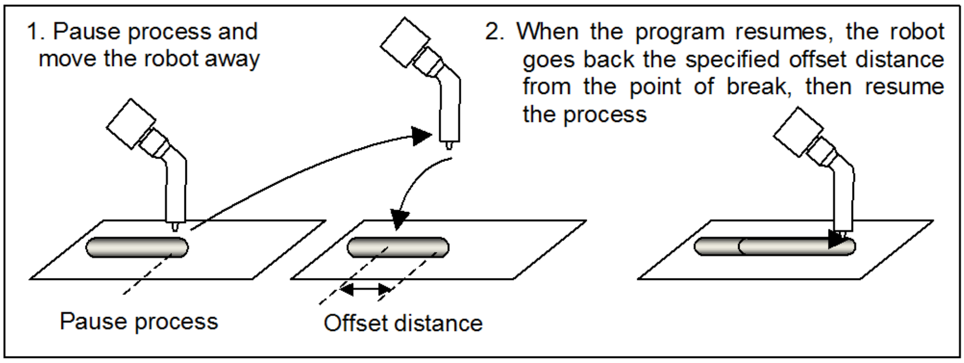


Fig. 9.13.2(a) Resume Offset

There are two types of Resume Offset, "Constant Path Resume Offset", and "Non-Constant Path Resume Offset". One type is automatically chosen according to the following table:

Table 9.13.2(a) Type of Resume Offset

Application Tool Software	Constant Path ($CPCFG.$CP_ENABLE)	Motion with Specific Option(*)?	Type of Resume Offset
Not Arc Tool	Enabled (TRUE)	No	Constant Path Resume Offset
		Yes	Non-Constant Path Resume Offset
	Disabled (FALSE)	-	Non-Constant Path Resume Offset
Arc Tool	-	-	Non-Constant Path Resume Offset

(*) Search Motion of Touch Sensing, Line Tracking, Weaving, and Servo Gun motion with pressurization

There are some special setup manners and restrictions specific to application software.

If Arc Tool, set up the Resume Offset in the dedicated manner.
If Dispense Function, set up the Resume Offset partially in the dedicated manner.
For details of setup manner, refer to the operator's manual of each application.

- Arc Welding Function OPERATOER'S MANUAL (B-83284EN-3) : "Weld Restart Function" in "3.3 ARC WELD SYSTEM SETUP" .
- Dispense Function OPERATOR'S MANUAL (B-83284EN-5) : "2.6 LIMITATIONS".

If Arc Tool, you can use Resume Offset only in Arc Welding process.
If Dispense Function, you can use Resume Offset only in Sealing process.

Limitations and usage notes of Resume Offset

- Resume Offset requires Original Path Resume. Resume Offset is subject to the limitations of Original Path Resume.
- Resume Offset does not work during single step execution.

- Resume Offset does not work for non-robot group (Independent Axes, Servo Gun, Positioners, etc.).
- Resume Offset uses the same Restart Move as Original Path Resume. The Restart Move goes directly to the Resume Offset stop position, not backward along the original path. Therefore, make sure there is no obstacle along the Restart Move.
- The actual Resume Offset position will be close to the desired offset, but usually not exact.
- In addition, there are some limitations and usage notes of Constant Path Resume Offset and Non-Constant Path Resume Offset respectively. Refer to the corresponding subsection for details.

Constant Path Resume Offset

Overview

Constant Path Resume Offset works in conjunction with the Constant Path option.
Constant Path Resume Offset places “offset stop position” at the previous position of the path when TP program is paused. When the TP program is resumed, the robot will first execute a Restart Move to the offset stop position. Next, the Resume Move will accelerate from the offset stop position, follow the identical path to the original stop position, and then continue execution normally.
Fig.9.13.2(b) shows how the Resume Offset works when the TP program is resumed after HOLD and jog feed. Even if there was no jog feed, the Restart Move goes to the same offset stop position.

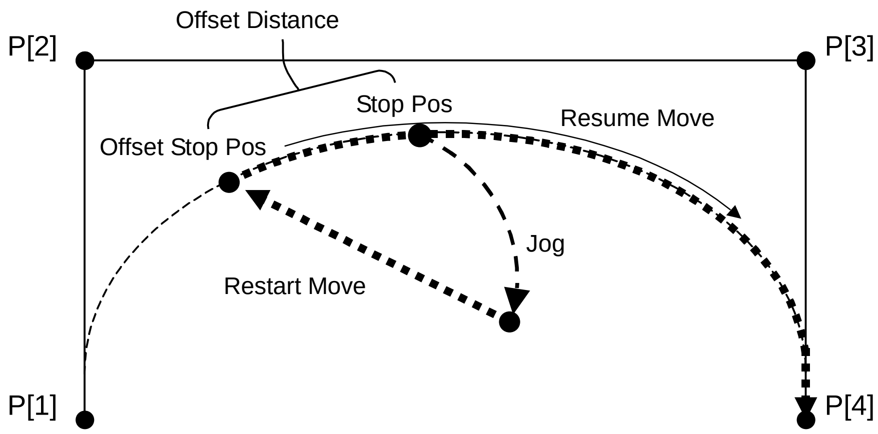


Fig. 9.13.2 (b) Constant Path Resume Offset

Setting of Constant Path Resume Offset

The Resume Offset Setup Menu is used to configure the Constant Path Resume Offset. This menu can be found in MENU -> SETUP -> Resume Offset.

```
Resume Offset Menu
                                         1/3
      1  Resume Offset Status:     DISABLE
      2  Max. Res. Offset Dist:      0.00mm
      3  Nominal Process Speed:      0.00mm/s

   [ TYPE ]   HELP    DONE    ENABLE   DISABLE
```

Fig. 9.13.2 (c) Resume Offset Menu

The Table. 9.13.2 (b) lists and explains the Resume Offset Setup Menu items.

Table 9.13.2(b) Resume Offset Setup Menu Items

Item	Description
Resume Offset Status	This item indicates whether Constant Path Resume Offset is enabled.
Max. Res. Offset Dist	This item is the maximum resume offset you expect to use. You need to set an actual offset distance as described below. Here, enter the maximum actual offset distance into the this item.

Item	Description
Nominal Process Speed	This item is the nominal resume offset speed: the program speed at which you want your process to execute. In other words, the taught speed of the process in the TP program. If various speed is expected in the TP program, enter the minimum expected speed.

When you change the value of each item, a message "Press F3(DONE) to finish setup" is displayed.
After you enter values in all items, press F3(DONE). Then, the following popup will be displayed. (The same popup will be displayed when you leave this setup screen.)
If you select "YES", the controller will automatically cycle its power. After that, the change is effective.
If you select "NO", the change will be cancelled.

```
Is it OK to cycle power now?

Please note change required
cycling power to take effect
Answer NO will remove change

     [ YES ]         NO
```

NOTE
If your controller is R-30*i*B Mate, do manual power cycle after you select "YES".
If the software series is 7DD0 (V8.13P), this popup is not displayed. In this case, do manual power cycle two times.

After setup in the menu screen is finished, set the offset distance.

- If Dispense Function, you must set the offset distance in the dedicated manner. Refer to the Dispense Function operator's manual
- If other application software, you can set the offset distance using the following system variable:
  $MCR_GRP[group no.].$RSM_OFFSET : Offset distance in mm (>=0)
  When this value is 0, Resume Offset is disabled.
  You need to change this system variable not in system variable screen, but in TP programs. For details, refer to the subsection "Programming" described below.

Limitations and usage notes of Constant Path Resume Offset

- Constant Path Resume Offset requires the Constant Path option. If Constant Path is disabled, Non-Constant Path Resume Offset is effective. Note that depending upon software configuration, Non-Constant Path Resume Offset can be effective even if Constant Path is enabled, as described above.
- Constant Path Resume Offset requires the offset distance to be set before the robot has stopped due to HOLD etc. Changing the offset distance after HOLD but before Resume has no effect.
- Constant Path Resume Offset can't offset past the beginning of the last motion sequence (the start position of the program, or the taught position with FINE or CNT0 positioning path). Attempting to do so will only move to the beginning of the last motion sequence.
- Constant Path Resume Offset can't offset beyond Max. Res. Offset Dist. Attempting to do so will only offset the Max. Res. Offset Dist.
- Max. Res. Offset Dist in the setup menu does not exactly mean the maximum offset distance The actual maximum offset distance can be calculated as below:
  Actual Max. Offset Dist = Setup value in Max. Res. Offset Dist
  / Setup value in Nominal Process Speed * Actual motion speed

For example, if the setup value in Nominal Process Speed is slower than the actual speed, the actual maximum offset distance is smaller than setup value. In order to avoid this, enter the smallest expected taught speed into Nominal Process Speed.

- Constant Path Resume Offset can't be used in the TP program which involves multiple motion groups. For such TP programs, set offset distance ($MCR_GRP[].$RSM_OFFSET) of every motion group to 0. Otherwise, "CPMO-004 Feature not Supported" alarm will be posted every time the TP program pauses.

Non-Constant Path Resume Offset

Non-Constant Path Resume Offset works without the Constant Path option.
Non-Constant Path Resume Offset places "offset stop position" along the line between "the original stop position" and "the last past taught point" when TP program is paused. When the TP program is resumed, the robot will first execute a Restart Move to the offset stop position. Next, the Resume Move will accelerate from the offset stop position to perform the rest of the program.
Fig.9.13.2(d) shows how the Non-Resume Offset works when the TP program is resumed after E-Stop and jog feed. Even if there was no jog feed, the Restart Move goes to the same offset stop position.

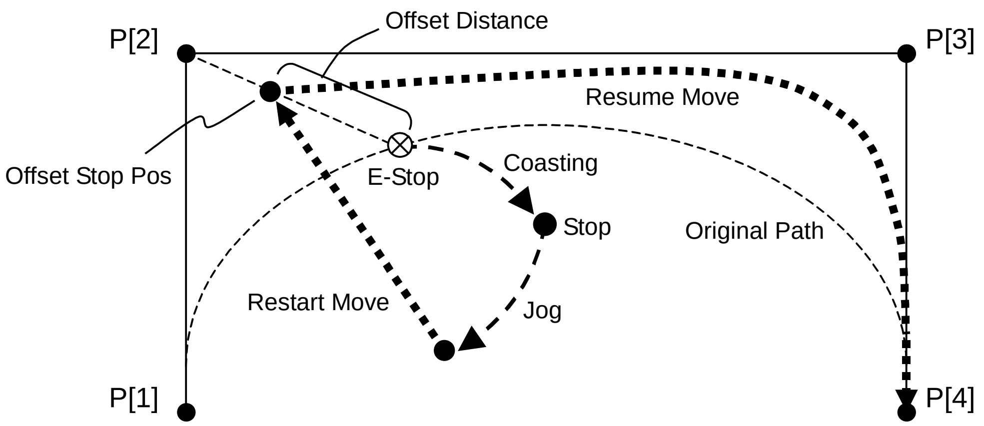


Fig. 9.13.2(d) Non-Constant Path Resume Offset

The Resume Move path is generally not on the original path, but will be close for applications with slow program speeds or limited corner blending.

Setting of Non-Constant Path Resume Offset

- If Arc Tool, set up in the dedicated manner. Refer to the operator's manual of Arc Tool.
- If Dispense Function, set up the offset distance in the dedicated manner. Refer to the operator's manual of Dispense Function.
- If other application software, you can set the offset distance using the following system variable:
  When this value is 0, Resume Offset is disabled.
  You need to change this system variable not in system variable screen, but in TP programs. For details, refer to the subsection "Programming" described below.

Limitations and usage notes of Non-Constant Path Resume Offset

- Non-Constant Path Resume Offset can't offset before "the last past taught point". Attempting to do so will only move to "the last past taught point".
- Non-Constant Path Resume does not work for Circular motion or Circle arc motion.

Programming

To use Resume Offset, you need to set offset distance ($MCR_GRP[group no.].$RSM_OFFSET) correctly in the TP program. This subsection explains how to do it.

> **NOTE**
> If you application software is Arc Tool or Dispense Function, you must set offset distance in the dedicated manner. You don't need programming described in this subsection. For details, refer to the operator's manual of each application software.

In general, we need Resume Offset only during the process (such as cutting, welding, or dispensing) is ON. Therefore, we should set a certain offset distance at the beginning of the process, then reset to 0 at the end of the process. To set offset distance, use parameter instruction in the TP program.

Fig. 9.12.2(e) shows an example of program which uses Resume Offset.
Note that in actual system, you may need an automatic process OFF/ON control at pause/resume. In such a case, you may need PLC etc.

```
TEST
                                              10/10
        1: J P[1] 30% FINE
        2:    $MCR_GRP[1].$RSM_OFFSET=10      Set offset distance = 10mm
        3:    DO[1]=ON                        Turn ON process
        4: L P[2] 100cm/min CNT50             :
        5: L P[3] 100cm/min CNT50             : (Process is ON)
        6: L P[4] 100cm/min FINE              :
        7:    DO[1]=OFF                       Turn OFF Process
        8:    $MCR_GRP[1].$RSM_OFFSET=0       Set offset distance = 0mm (Disable)
        9: J P[5] 30% FINE
     [End]
```

Fig. 9.13.2(e) Example of program for Resume Offset

> **NOTE**
> Even if you use Resume Offset in all motions regardless of process OFF/ON, please set the offset distance using parameter instruction in the beginning of TP program, do not change the value via system variable screen.

9.14 MULTITASKING FUNCTION

9.14.1 Overview

The multitasking function enables multiple programs to be executed at the same time. A "task" refers to an active program. For example, the multitasking function allows the robot to work based on two programs being concurrently executed, one for controlling the robot and the other for controlling peripherals and extended axis (for multiple groups). Through concurrent execution of programs, the cycle time can be reduced, and the state (of input signals, for example) can be monitored during robot operation.

9.14.2 Functions

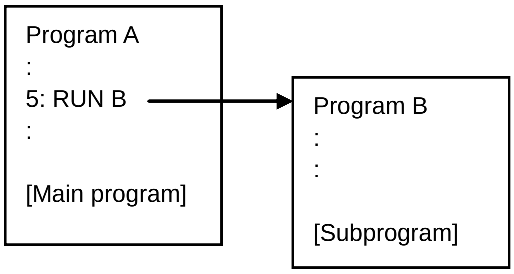
9.14.2.1 Notes on program creation

Create programs as usual. Note the following:

- A signal control program and a program intended only to read data do not use an operation group. Be sure to disable operation groups for these programs.
  On the program detail screen, a group mask should be set to [*,*,*,*,*,*,*,*].
- Programs cannot be executed at the same time if they use the same operation group.
- Programs that they use different operation groups can be executed at the same time.

9.14.2.2 How to start programs in multitasking

One program can start another as a task using the "RUN" instruction. The starter is referred to as a main program, while the started program is referred to as a subprogram.



In the above example, program A starts program B using the "RUN" instruction, and then both programs are executed concurrently. Program A is the main program of program B, and program B is the subprogram of program A.

If program A uses the "RUN" instruction to start program B as in the above example when program B is already active, an alarm is issued, suspending the execution of program A. In this case, program B must be terminated before program A is started.

9.14.2.3 Behavior of main program and subprogram

The main program and subprogram are halted

- When you select the main program and resume the execution, the execution of the subprogram is also resumed.
- When you select the subprogram and resume the execution, only the execution of the subprogram is resumed.
- When you select the main program and start backward execution, the subprogram is also executed in the backward direction.
- When you select the subprogram and start backward execution, only the subprogram is executed in the backward direction.

The main program is active and the subprogram is halted

- You cannot select the main program and resume the execution (in the forward or backward direction) (because the program is already active).
- When you select the subprogram and resume the execution, only the execution of the subprogram is resumed. This operation does not affect the main program execution.
- When you select the subprogram and start backward execution, only the subprogram is executed in the backward direction.

The main program is halted and the subprogram is active

- When you select the main program and resume the execution, the execution of the main program is resumed. The ongoing execution of the subprogram continues.
- When you select the subprogram and resume the execution, the ongoing execution of the subprogram continues. The execution of the main program is not resumed.
- When you select the main program and start backward execution, the main program is executed in the backward direction. The ongoing execution of the subprogram continues.
- When you select the subprogram and start backward execution, the subprogram is not executed in the backward direction. The ongoing execution continues. The main program is not executed in the backward direction.

Execution in single-step mode

- When you execute the main program in single-step mode, the subprogram is also executed in single-step mode.
- When you select the subprogram and resume the execution in single-step mode, only the subprogram is executed in single-step mode.

Halt and forced termination of program execution

- The main program and subprogram are independent of each other for halt and forced termination of program execution.
- Halting or forcedly terminating the main program does not affect the execution of the subprogram.

Execution of the main program in the backward direction

- After the execution of the main program is started in the backward direction, the execution stops at the "RUN" instruction and does not continue further.
- If you want the execution of the main program to continue in the backward direction beyond the "RUN" instruction, move the cursor to the line immediately before the "RUN" instruction.

9.14.2.4 Notes on main program and subprogram

Note the following when using the "RUN" instruction to start a subprogram:

- When the main program is executed in the forward or backward direction, the subprogram is also always executed in the same direction as the main program.
- If you want to execute the main program alone in the forward or backward direction, terminate the subprogram on the monitor screen, and then start execution in the forward or backward direction.
- If you want to execute the subprogram alone in the forward or backward direction, select the subprogram, and then start execution in the forward or backward direction. You do not need to terminate the main program.
- During execution in the backward direction, only motion instructions are executed. If synchronization between the main program and subprogram has been achieved by a register, the synchronization is not provided during the execution in the backward direction; the main program and subprogram are executed independently in the backward direction.
- If you want to run a subprogram after finishing another subprogram, please use register instruction and wait instruction as follows. In following example, wait 0.01 sec is taught in line 7 of Program A. This wait instruction is needed because it takes a little time for status of Program B to be finished after R[1] becomes to 1. If you remove this wait instruction, [PROG-040 Already locked by other task] may be posted when line 8 of Program A is executed.

```
Program A
:
:
4: R[1] = 0;
5: RUN B;          --->
6: WAIT R[1] = 1;
7: WAIT 0.01 (sec);
8: RUN C;
:
```

```
Program B
:
:
R[1] = 1;
[End]
```

9.14.3 Monitor

On the monitor screen, you can see at a time the state of multiple programs being executed. The active programs and halted programs are listed on the monitor screen.
The monitor screen shows the following information:

- Names of active or halted programs
- Names of active or halted subprogram (“CALL” instruction)
- Execution state (active, halted)

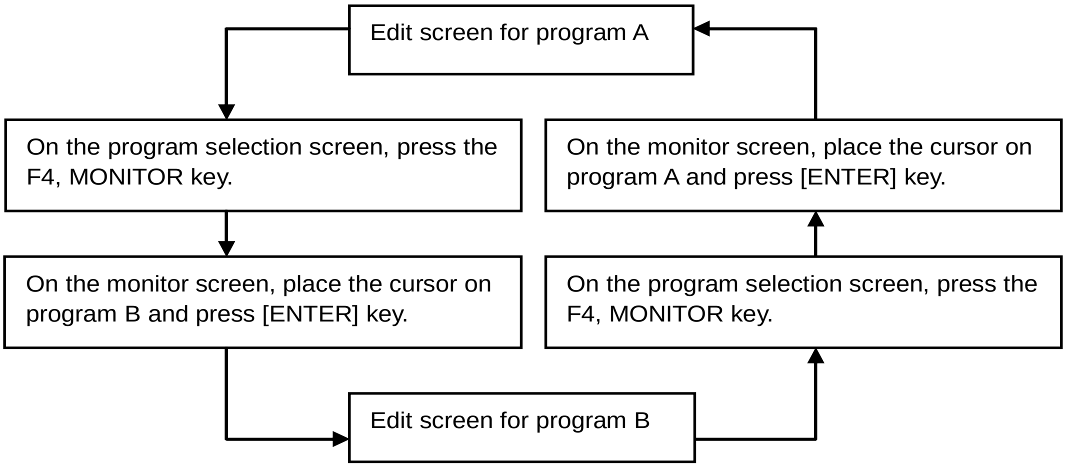
The monitor screen appears when you press the F4, MONITOR key on the program selection screen.

```
Select
               1014788 bytes free    1/10
     No.  Program name       Comment
      1  -BCKEDT-            [                 ]
      2  GETDATA          MR [Get PC Data      ]
      3  REQMENU          MR [Request PC Menu  ]
      4  SENDDATA         MR [Send PC Data     ]
      5  SENDEVNT         MR [Send PC Event    ]
      6  SENDSYSV         MR [Send PC Sysvar   ]
      7  PNS0001             [                 ]
      8  PNS0101             [                 ]
      9  PNS0002             [                 ]

   [ TYPE ]  CREATE   DELETE  MONITOR  [ATTR ]  >
```

```
Select (monitor)
                TASK           STATUS
    1 EXEC1    PNS0001   [PNS0001 :Running ]
    2 EXEC2    PNS0101   [DOOR_OP :Running ]


    Select monitor program
    PROG    ROUT    PAUSE   ABORT
```

Name of active program
Name of active subprogram (“CALL” instruction)
Program execution state

Place the cursor on a program you want to monitor, and press the [ENTER] key. The edit screen for the program appears, showing the execution state.
When multiple programs are active and you want to see the editing screen for each program in turn, you can switch among the screens on the above monitor screen with relative ease.
For example, when programs A and B are executed at the same time and you want to check the execution state of the programs on their respective edit screens, you should do the following:



When only one program is executed, pressing the F4, MONITOR key on the program selection screen skips the monitor screen and takes you directly to the editing screen. The monitor screen appears only when two or more programs are active or halted.

9.14.4 Halt and Forced Termination

If you do one of the followings while multiple programs are being executed, all active programs are halted:

- Press the hold button on the teach pendant or operator panel.
- Press the emergency stop button on the teach pendant or operator panel.
- Teach pendant is enabled when programs are started and executed by anything but teach pendant.
- Teach pendant is disabled, release [SHIFT] key, or release the deadman switch when programs are started and executed by teach pendant.
- Turn off the immediate stop signal (*IMSTP), halt signal (*HOLD), safety speed signal (*SFSPD), or operation enable signal (*ENBL).

When you select ABORT from the function menu, all active or halted programs are forcedly terminated.
Among the multiple programs being executed, you can halt or forcedly terminate only the specified program on the monitor screen, as described below.

Press the F4, MONITOR key on the program selection screen. The monitor screen appears. Place the cursor on a program you want to halt or forcedly terminate and press the F3, PAUSE key or F4, ABORT key. Only the specified program is halted or forcedly terminated.

```
Select (monitor)
                TASK            STATUS
   1 EXEC1    PNS0001   [PNS0001 :Running ]
   2 EXEC2    PNS0101   [DOOR_OP :Runinng ]


   Select monitor program
 | PROG | ROUT | PAUSE | ABORT |  |
```

```
Select (monitor)
                TASK            STATUS
   1 EXEC1    PNS0001   [PNS0001 :Runinng ]
   2 EXEC2    PNS0101   [DOOR_OP :PAUSE   ]


   Select monitor program
 | PROG | ROUT | PAUSE | ABORT |  |
```

9.14.5 Cycle Stop Signal (CSTOPI)

The cycle stop signal can be set to forcedly terminate active or halted programs externally. This setting can be made on the system setting screen.
Use “6 SYSTEM” to display the system setting screen. When you set "CSTOPI for ABORT" to "TRUE", the currently selected program is terminated by the cycle stop signal. When you set "Abort all program by CSTOPI" to "TRUE", all programs can be forcedly terminated.

9.15 ERROR SEVERITY TABLE

9.15.1 Error Severity Table Overview

The Error Severity Table screen allows you to modify severity levels for error codes in the direction of a higher severity level. In addition, you can output an error code to a PLC system with a severity level of PLCWARN, or use miscellaneous error logger functions.

9.15.2 Modifying Error Severity

When the system is about to display an error message, it first checks the severity level set in the system variable $ERROR_TABLE[ ] and compares the level with the default setting. If the severity level is higher than the default setting, the error code is then posted with the modified severity level setting. The system also checks the error logger functionality and logs the error accordingly.
When software is installed for the first time on the controller, the Error Severity Table screen looks similar to Error Severity Table (Fig.9.15.2).

```
Error Severity
    Error Severity Table               1/20
      # FCode FName ECode   Sever   Erlog
      1  ****  ****     0  DEFAULT  DEFAULT
      2  ****  ****     0  DEFAULT  DEFAULT
      3  ****  ****     0  DEFAULT  DEFAULT
      4  ****  ****     0  DEFAULT  DEFAULT
      5  ****  ****     0  DEFAULT  DEFAULT
      6  ****  ****     0  DEFAULT  DEFAULT
      7  ****  ****     0  DEFAULT  DEFAULT
      8  ****  ****     0  DEFAULT  DEFAULT
      9  ****  ****     0  DEFAULT  DEFAULT
     10  ****  ****     0  DEFAULT  DEFAULT

   [ TYPE ]  VERIFY  CLEAR          HELP
```

Fig. 9.15.2 Error severity table

Memo

The default number of entries is 20, but can be extended to 999. Refer to Table 9.15.2 for a listing and description of the items on the severity table that you can modify or edit for each kind of error.

Table 9.15.2 Error severity table items

Item	Description
FCode (Facility Code) and FName	This item is a decimal number which represents the subsystem in which an error has occurred. When you enter a number for the facility code, the system checks whether the number is a valid facility code. If not, an error message is displayed and you are prompted to enter a valid number. If the number is valid, the Facility Name will be displayed in the FName field of the table. Refer to for a complete listing of facility names and codes.
ECode (Error Code)	This item is a decimal number which is used by the system to report the proper error on the error message line of the teach pendant screen. Refer to for a complete listing of error codes and descriptions.
Sever (Severity Level)	This item indicates how serious the error is. This item can be modified, but only towards a higher severity level. For each possible error, you can set the following severity levels: • DEFAULT - This setting means that the error table has no effect on severity level for this error, and the default value of severity will be displayed. If the error output option is loaded, error codes with DEFAULT error severity levels will be output to the PLC system if the severity level is higher than WARNING. • STOP - This setting pauses program execution and stops robot motion. When a motion is stopped, the robot decelerates to a stop and any remaining part of the current motion segment is saved, meaning that motion can be resumed. STOP errors usually indicate that some action must be taken before the motion and program execution can be resumed. With these kinds of errors, the operator panel FAULT light and the teach pendant FAULT LED will be turned on. This severity level is for local tasks only. • STOPALL - This setting pauses program execution and stops robot motion. This severity level is for all tasks. • ABORT - This setting aborts program execution and stops robot motion. When an abort error occurs, the robot decelerates to a stop and the remainder of the motion segment is cancelled. This severity level is for local tasks only.

Item	Description
Sever (Severity Level)	• ABORTALL - This setting aborts program execution and stops robot motion. This severity level is for all tasks. • PLCWARN - If the error code output option is loaded, error codes with PLCWARN severity levels will be output to the PLC system with global severity bits set. In order to use this output option, you must have 33 digital outputs and one digital input which can be dedicated to this option. Refer to for information about setting up the error code output option.
Erlog (Error Logger)	This item provides different actions for each possible error. For each possible error, you can choose one of the following: • DEFAULT - This setting is the default setting. • ACTIVE - This setting displays an alarm in the active alarm log. • NODISP - This setting indicates that an error will not be displayed in either the error log or on the error line. • NOERLOG - This setting indicates that an error will not be displayed in the error log. • NOERLIN - This setting indicates that an error will not be displayed on the error line.

Use Procedure Modifying the Error Severity Table (Procedure9-29) to modify the error severity table. Use Procedure Modifying the Number of Entries in the Error Severity Table (Procedure9-30) to modify the number of entries in the error severity table.

Procedure 9-29 Modifying the error severity table

Conditions

- You need to know the decimal numbers for the facility code and the error code so that you can modify the ERROR SEVERITY setup screen.

Steps

1 Press [MENU] key.
2 Select SETUP.
3 Press F1, [TYPE].
4 Select NEXT, then select Error Table. The following screen will be displayed.

```
Error Severity
    Error Severity Table              1/20
      # FCode FName ECode  Sever   Erlog
      1    11   SRVO    6  DEFAULT  DEFAULT
      2    11   SRVO    4  DEFAULT  DEFAULT
      3    10   FLPY    2  DEFAULT  DEFAULT
      4    10   FLPY    3  DEFAULT  DEFAULT
      5     3   PROG    5  DEFAULT  DEFAULT
      6     3   PROG    6  DEFAULT  DEFAULT
      7     2   FILE    1  DEFAULT  DEFAULT
      8     2   FILE    2  DEFAULT  DEFAULT
      9    33   DICT    8  DEFAULT  DEFAULT
     10  ****  *****    0  DEFAULT  DEFAULT

   [ TYPE ]  VERIFY   CLEAR          HELP
```

5 If you want to add or modify an entry:

a. Move the cursor to the FCode field for the row you want to edit and type a decimal number. If the number is not valid, the error message "Invalid Facility Code entered" will be displayed. After you type a valid number, the FName field will automatically display the new facility code that corresponds to the new FCode.

b. Move the cursor to the ECode field and type a decimal number within the range of valid numbers (0 -999). If the number is not valid, the error message "Invalid integer (0 -999)" will be displayed.

c. Move the cursor to the Sever field and press F4, [CHOICE]. The following screen will be displayed.

```
                 1
1 DEFAULT
2 STOP
3 STOPALL
4 ABORT
5 ABORTALL
6 PLCWARN
7
8
```

d. Move the cursor to the desired severity level and press the [ENTER] key.
The PLCWARN choice is only available if the system variable $ER_OUT_PUT.$PLCWARN is set to TRUE. The default setting for this variable is FALSE.

e. To modify Error Logger functionality, move the cursor to the Erlog field and press F4, [CHOICE]. The following screen will be displayed.

```
                 1
1 DEFAULT
2 ACTIVE
3 NODISP
4 NOERLOG
5 NOERLIN
6
7
8
```

f. Move the cursor to the desired error logger functionality and press the [ENTER] key.

g. To verify that the proper error code was entered, press F2, VERIFY.

h. To display help information, press F5, HELP. When you are finished, press PREV to return to the Error Severity Table screen.

i. To clear information from a row, move the cursor to the row whose information you want to clear, and press F3, CLEAR. The message "Clear this entry?" will be displayed.
Press F4, YES, to clear the current row.

Procedure 9-30 Modifying the number of entries in the error severity table

Steps

1 Turn on the controller with [PREV] and the [NEXT] key pressed. Then select [3 Controlled start].
2 Press [MENU] key
3 Press 0, NEXT, and select Program Setup. The following screen will be displayed.

```
Program Limits
      Program Limits Setup              1/13

      1  User Tasks                    4
      2  Numeric Registers             200
      3  Position Registers            100
      4  String Registers              25
      5  Pallet Registers              32
      6  Macros                        150
      7  User Alarms                   10
      8  Trace Length                  200
      9  Num. Dig. Ports               512
     10  Error Severity Table          20

   [ TYPE ]                          HELP
```

4 Move the cursor to Error Severity Table, type the desired number of entries (1 -999), and press the [ENTER] key.
5 Press [FCTN] key.
6 Select Start (Cold).

9.16 DIAGNOSTIC LOG

9.16.1 Overview

This function is to save the internal data of the robot controller by very easy operation just after some problems are occurred. Some internal data are removed by power off. After saving its data into FROM by this function, please send the image backup to FANUC. Then problem investigation could be started.

NOTE
It is fine to execute image backup after it is allowed to do cycle power. It is not needed to do just after having done diagnostic log. However it would be very nice if you could send FANUC to the backup as soon as possible for a prompt investigation.

9.16.2 Operations

NOTE
- Do the following operation just after the problem occurs and BEFORE POWER OFF.
- Do it only one time just after the problem occurs.
  The save data is deleted if the same operation is done twice time.

⚠ WARNING
If you execute this function, confirm that the status of robot is STOP. In some cases the robot becomes slow motion while saving log.

Select “Diagnostic log” in FCTN menu when the problem is occurred. The operational procedure is as follows. Refer to “SAVING DIAGNOSTIC DATA” in the APPENDIX too.

1 Press [FCTN] key on TP and select “0 – Next –“ → “0 –Next –“ → “7 Diagnostic log”.
2 “Diagnostic log” is displayed on 3rd page in FCTN menu. (See following sample figure.)

```
UTILITIES Hints
                                   FUNCTION 3
               HandlingToo      1 REFRESH PANE
       V8.10P/01                2
                                3
                                4
    Copyright 2012, All Righ    5
            FANUC CORPORATIO    6
     FANUC Robotics America     7 Diagnostic log
   Licensed Software: Your u    8 Del Diag Log
   Your acceptance. This pro    9
   By several U.S. patents.     0 -- NEXT --

   [ TYPE ]  LICENSE  PATENTS          HELP
```

3 Set the cursor onto “7 Diagnostic log” and press the [ENTER] key.

4 It takes about 1 minute to save diagnostic log.
5 "SYST-274 Diagnostic log done" is posted after diagnostic log is done. If this warning is not posted you can confirm by disappearing the FCTN menu.

No need to insert an external memory device such as memory card and USB because data is saved to FROM.

If the following message is displayed, please execute image backup and delete the data saved in FROM from FCTN menu. How to delete is described later, the subsection "Delete the Data Saved in FROM".

```
FROM has low free space.
Execute image backup and
Delete diagnostic log.


          [ OK ]
```

9.16.3 Getting Out the Save Data

Please get out saved data to an external device such as MC: and so on by either one of the following 2 methods depending on the situation.

If it allows to do cycle power, please execute image backup.

Saved data is output by executing image backup. Refer to the section "IMAGE BACKUP FUNCTION" in the chapter "FILE INPUT/OUTPUT" to know more.

If not, please save maintenance data.

In the case image backup cannot be executed because cycle power is not allowed, save the maintenance data from FILE screen. Refer to "Saving all the program Files Using the File screen" in the chapter "FILE INPUT/OUTPUT" to know more.

9.16.4 Output to External Device

It is not allowed to save the data of diagnostic log to FROM if lack of FROM. "SYST-273 Lack of FRA for saving" is posted. In this case it allows to output it to MC: or UD1:.

> **NOTE**
> You must use memory card (MC:) and USB memory (UD1:) formatted at the robot controller.

```
Lack of FROM space.
Save Diagnostic Log to
MC:. Are you sure?



    [ YES ]       NO
```

Insert the memory card to the robot controller and select YES. Diagnostic log is output to memory card. If NO is selected, confirming message to save to UD1: is displayed. Insert USB memory to the robot controller and select YES. Diagnostic log is output to USB memory.

In case of R-30*i*B Mate, R-30*i*B Mate Plus, R-30*i*B Compact Plus and R-30*i*B Mini Plus controllers, only confirming message saving to UD1: is displayed.

9.16.5 Delete the Data Saved in FROM

To delete saved data, execute following operation.
Press [FCTN] key on the teach pendant and select "0 – Next –" → "0 – Next –" → "8 Del Diag Log".

```
UTILITIES Hints
                                       FUNCTION 3
                  HandlingToo     1 REFRESH PANE
        V8.10P/01                 2
                                  3
                                  4
     Copyright 2012, All Righ     5
             FANUC CORPORATIO     6
      FANUC Robotics America      7 Diagnostic log
    Licensed Software: Your u     8 Del Diag Log
    Your acceptance. This pro     9
    By several U.S. patents.      0 -- NEXT --

   [ TYPE ]  LICENSE  PATENTS          HELP
```

Select YES after confirming message will be displayed. Then saved data will be deleted.

```
Delete Diagnostic Log?



    [ YES ]        NO
```

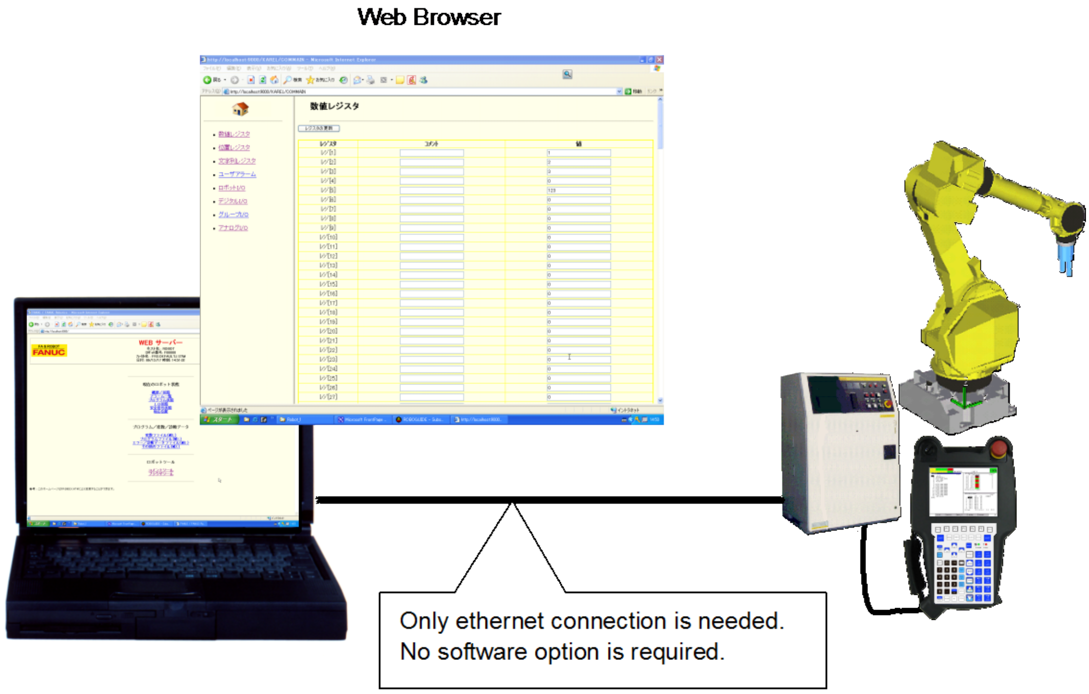
9.16.6 Limitation

Depending on the condition of occurring problem, this function is not available. For example, in the case TP is frozen and any key operation are ignored.

9.17 ROBOT TOOLS OF ROBOT HOMEPAGE

The web server allows you to access to robot home page. The links on the default home page called “ROBOT TOOLS” allow you to enter text information easily using a keyboard.

- PC requires only web browser(Internet explorer) to use ROBOT TOOLS.
- Robot controller does not need software option.

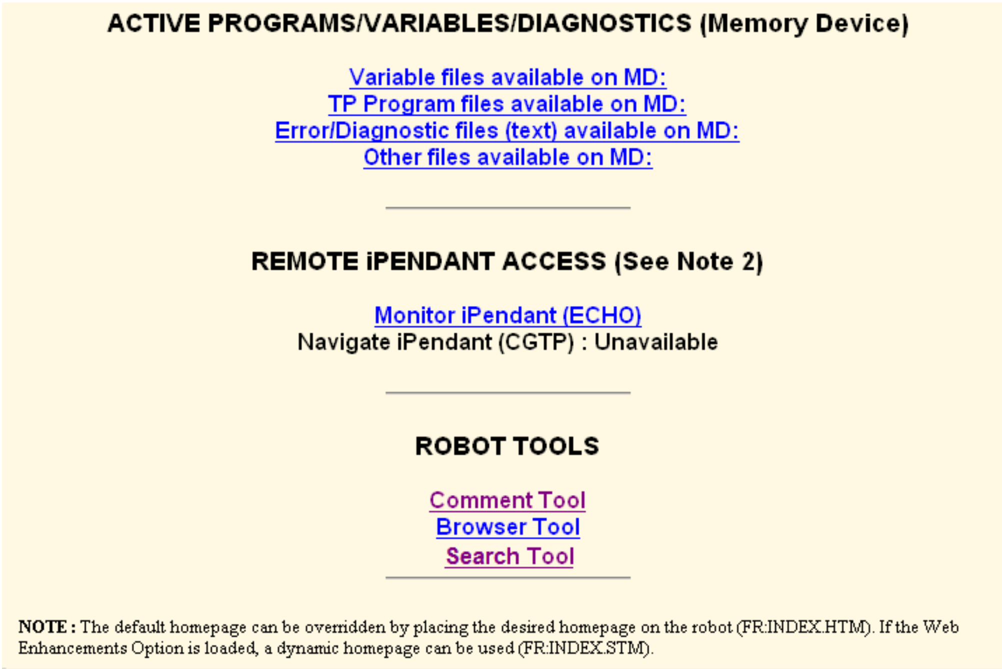


NOTE

1 Before using ROBOT TOOLS, you have to allow web server to run KAREL program. Otherwise controller requests you to input username and password even though you have not setup them. If you don’t need authentication, display HTTP AUTHENTICATION screen and set “KAREL” to “U” (Unlock). Please refer to 9.17.1.2.

2 You can display robot homepage by the teach pendant. In this case, ROBOT TOOLS are not displayed.

Following figure shows part of robot homepage, including ROBOT TOOLS.



ACTIVE PROGRAMS/VARIABLES/DIAGNOSTICS (Memory Device)

Variable files available on MD:
TP Program files available on MD:
Error/Diagnostic files (text) available on MD:
Other files available on MD:

REMOTE iPENDANT ACCESS (See Note 2)

Monitor iPendant (ECHO)
Navigate iPendant (CGTP) : Unavailable

ROBOT TOOLS

Comment Tool
Browser Tool
Search Tool

NOTE : The default homepage can be overridden by placing the desired homepage on the robot (FR:INDEX.HTM). If the Web Enhancements Option is loaded, a dynamic homepage can be used (FR:INDEX.STM).

This section first describes outlines of setup required to display ROBOT TOOLS by PC. Then ROBOT TOOLS are described.

9.17.1 Setup for ROBOT TOOLS

This Subsection explains setup of IP address and HTTP authentication.

9.17.1.1 Setup IP Address

1 Setup TCP/IP on robot controller.
(a) Press [MENU] key.
(b) Select SETUP.
(c) Press F1, [TYPE].
(d) Select Host Comm. The following screen will be displayed.

```
SETUP Protocols
                                              1/8
        Protocol    Description
      1 TCP/IP      TCP/IP Detailed Setup
      2 TELNET      Telnet Protocol
      3 SM          Socket Messaging Device
      4 PROXY       Proxy Server
      5 PPP         Point to Point Protocol
      6 PING        Ping Protocol
      7 HTTP        HTTP Authentication
      8 FTP         File Transfer Protocol

 [ TYPE ]          DETAIL   [ SHOW ]
```

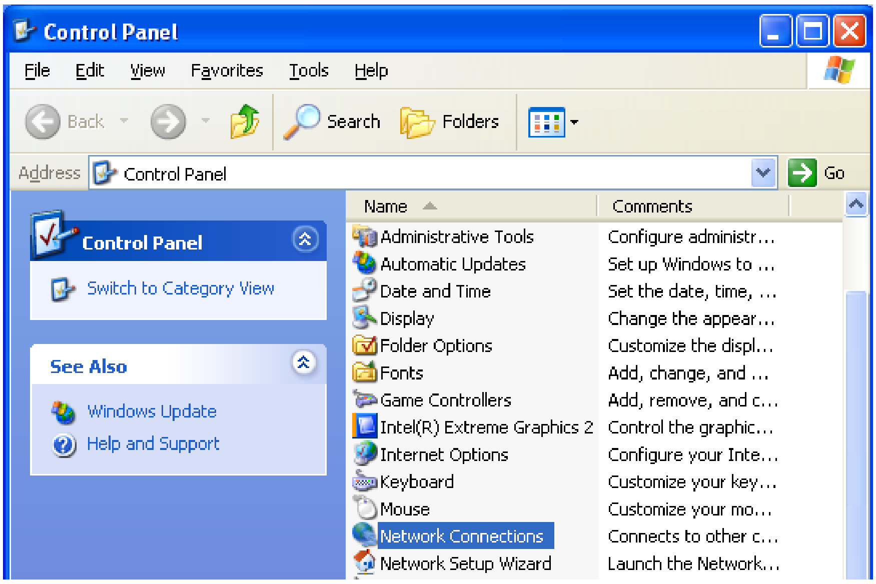
(e) Select TCP/IP and press F3, DETAIL. The following screen will be displayed.

```
SETUP Host Comm
    TCP/IP                                  3/40
        Robot name:                    HANDLING
        PORT#1 IP addr:             192.168.0.4
        Subnet Mask:                255.255.0.0
        Board address:    ******************
        Router IP addr:   ******************

      Host Name (LOCAL) Internet Address
        1 **********      ******************
        2 **********      ******************
        3 **********      ******************
        4 **********      ******************

   [ TYPE ]         PORT    PING    HELP    >
```

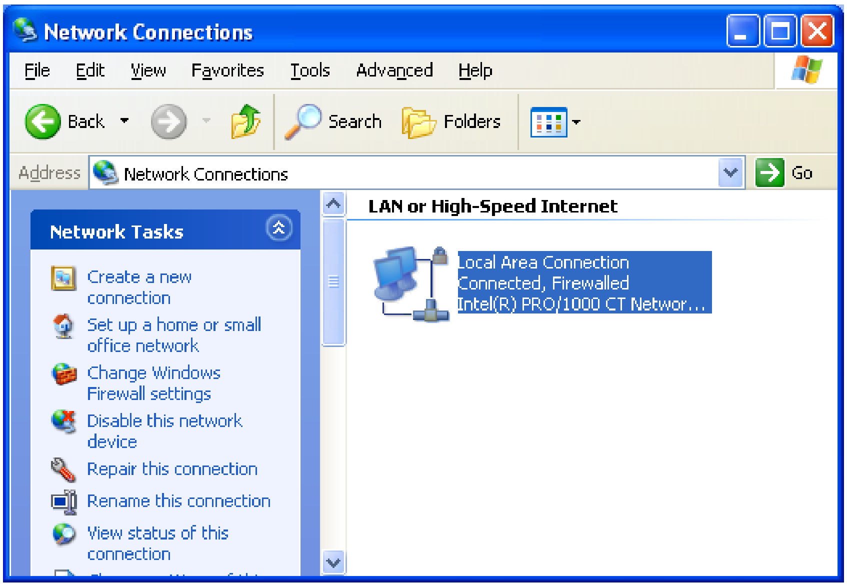
2 Set IP address.
Set Subnet Mask and Router IP address as required.

3 PC needs setup to communicate by TCP/IP.
If fixed IP address is available and Windows XP is installed to your PC, following procedure may help you. You have to setup TCP/IP network according to your environment.

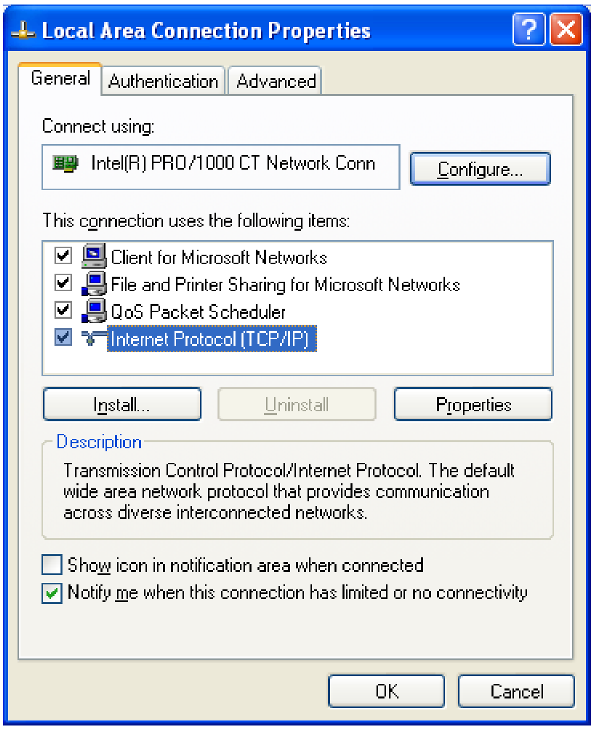
Select Network Connections of control panel.



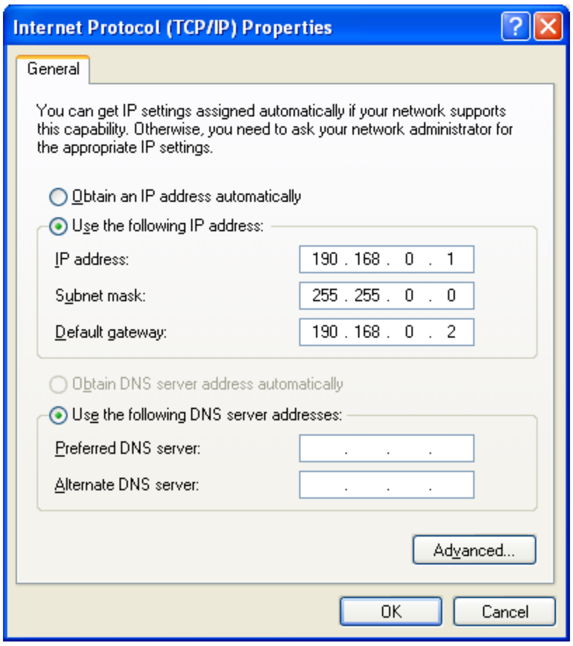
Select Local Area Connection and display property.



Select Internet Protocol (TCP/IP) and display property. Display IP address.



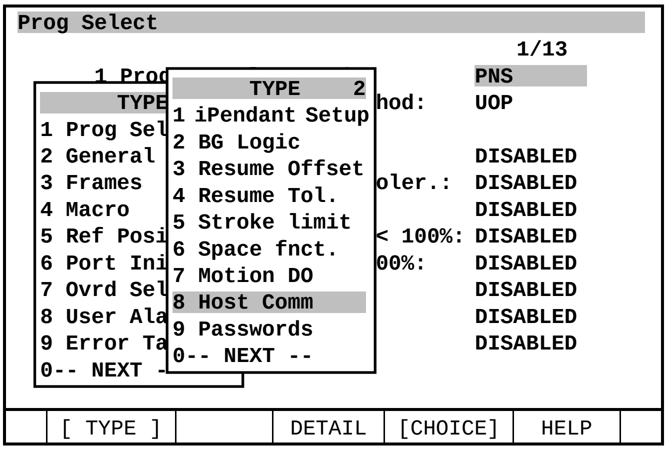
Input IP address.



9.17.1.2 Setup HTTP AUTHENTICATION

Before using ROBOT TOOLS, you have to allow web server to run KAREL program. Otherwise controller requests you to input username and password even though you have not setup them. If you don't need authentication, display HTTP AUTHENTICATION screen and set "KAREL" to "U" (Unlock).

1 Press [MENU] key.
2 Select SETUP.
3 Press F1, [TYPE] and select Host Comm.



4 Move cursor to HTTP and press F3, DETAIL.

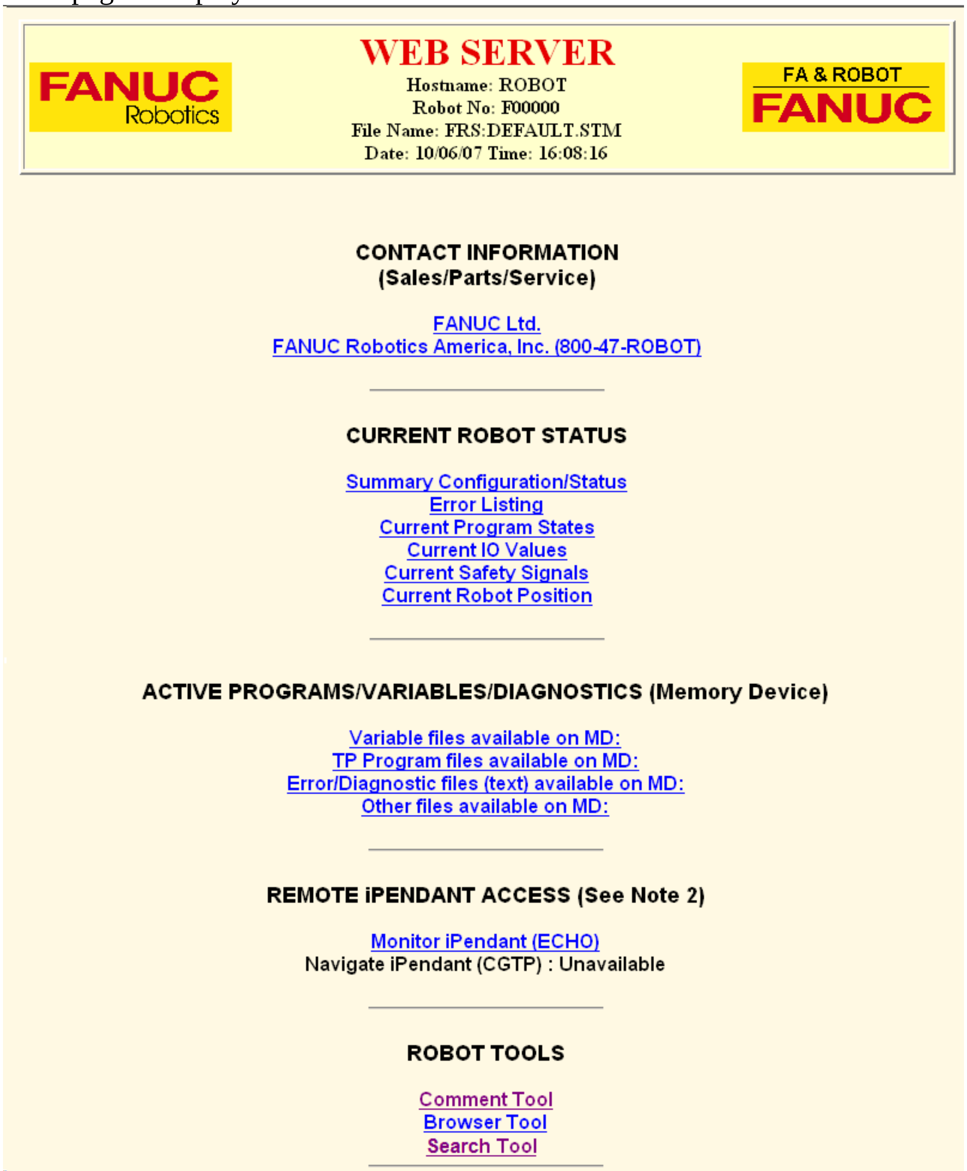
```
SETUP Protocols
                                              1/8
        Protocol    Description
      1 TCP/IP      TCP/IP Detailed Setup
      2 TELNET      Telnet Protocol
      3 SM          Socket Messaging Device
      4 PROXY       Proxy Server
      5 PPP         Point to Point Protocol
      6 PING        Ping Protocol
      7 HTTP        HTTP Authentication
      8 FTP         File Transfer Protocol

   [ TYPE ]       DETAIL  [ SHOW ]
```

5 Move cursor to head of “KAREL” line and press F3 UNLOCK.

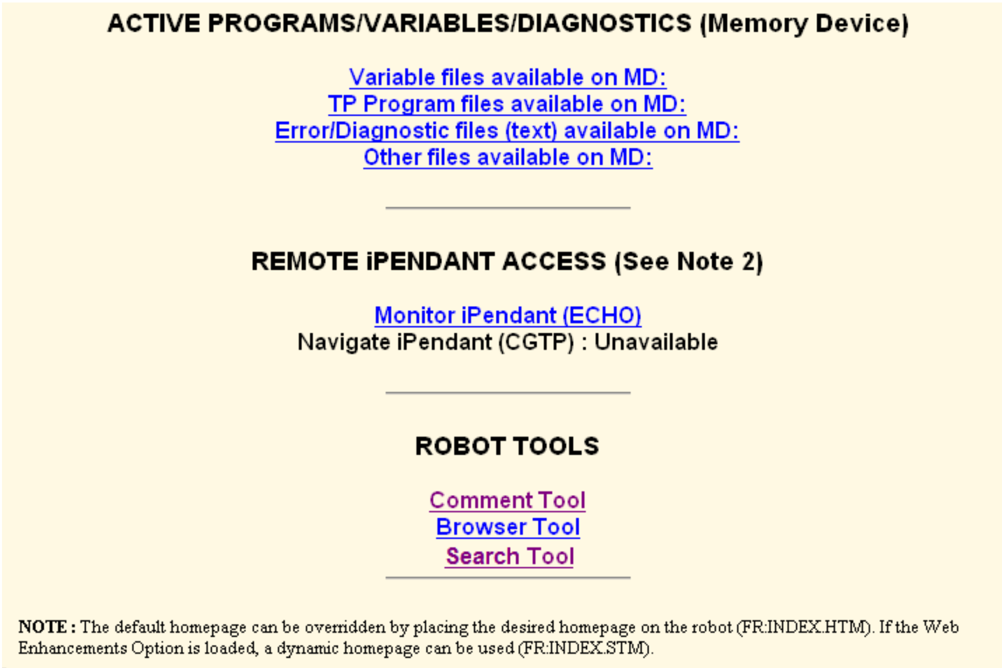
```
HTTP Setup
                                              2/8
             PROTECTED RESOURCES
       Name   Pwrd  Resource
    A ****** ****** iPendant
    U ****** ****** KAREL:*
    A ****** ****** KCL:*
    U ****** ****** VISION SETUP
    A ****** ****** ************************
    A ****** ****** ************************
    A ****** ****** ************************
    A ****** ****** ************************

   [ TYPE ]   LOCK   UNLOCK   AUTH   HELP
```

9.17.2 Usage of ROBOT TOOLS

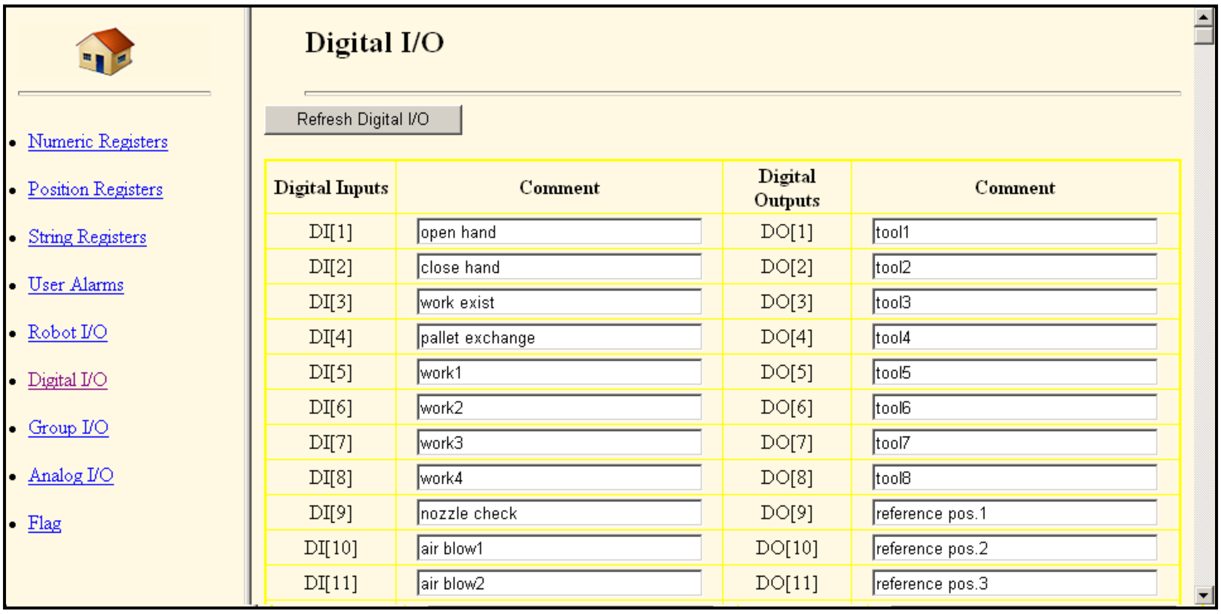
1 Invoke Internet Explorer.
2 Set “http://(IP address of ROBOT)/” to URL and display it.
Robot home page is displayed.



The links on the default home page called “ROBOT TOOLS” allow you to enter text information easily using a keyboard. In the ROBOT TOOLS, there are two links. They are Comment Tool and Browser Tool and Search Tool.



"Comment Tool" will split the screen as shown in following figure. The left side links determine the type of robot information displayed on the right side. The data is retrieved from the robot controller but it is not dynamically updated. You must press the "Refresh" button if you want to refresh the contents of the page. After entering text in a text box, click the mouse outside the box or press the tab key. The new data is immediately sent to the robot. When you wish to return home, press the HOME icon on the upper left of the screen.

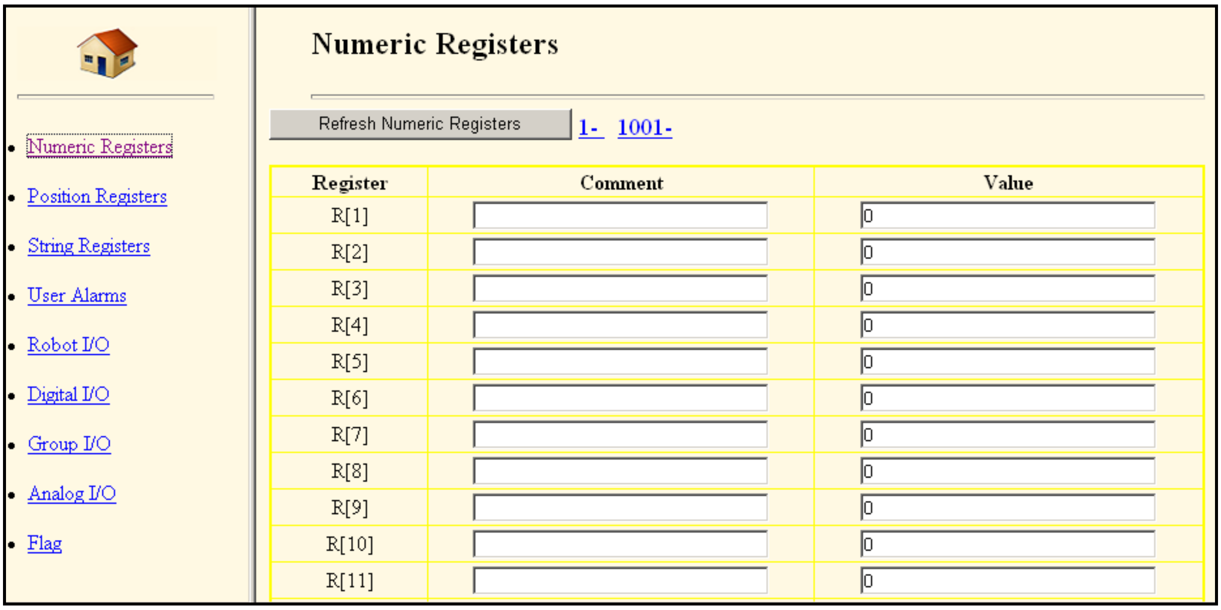


Digital Inputs	Comment	Digital Outputs	Comment
DI[1]	open hand	DO[1]	tool1
DI[2]	close hand	DO[2]	tool2
DI[3]	work exist	DO[3]	tool3
DI[4]	pallet exchange	DO[4]	tool4
DI[5]	work1	DO[5]	tool5
DI[6]	work2	DO[6]	tool6
DI[7]	work3	DO[7]	tool7
DI[8]	work4	DO[8]	tool8
DI[9]	nozzle check	DO[9]	reference pos.1
DI[10]	air blow1	DO[10]	reference pos.2
DI[11]	air blow2	DO[11]	reference pos.3

NOTE

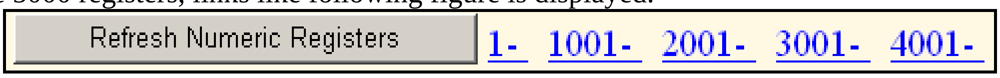
Flag is displayed on the left frame if system software version is later than 7DC1/12 or 7DD0/07. If system software version is series 7DC2, all versions displays link for Flag.

Software series 7DC2 supports optional software, Expanded registers function (R830). The number of registers and position registers can be more than 1000 if the option is ordered. If there are more than 1000 registers, click of link of register in left frame displays from R [1] to R [1000] on the right frame.



Links to show range of index is displayed near the beginning of the right frame. If there are 1500 registers, links like following figure will be displayed. If you click “1-“, right frame shows from R [1] to R [1000]. Click of “1000-“ shows from R [1001] to R [1500].

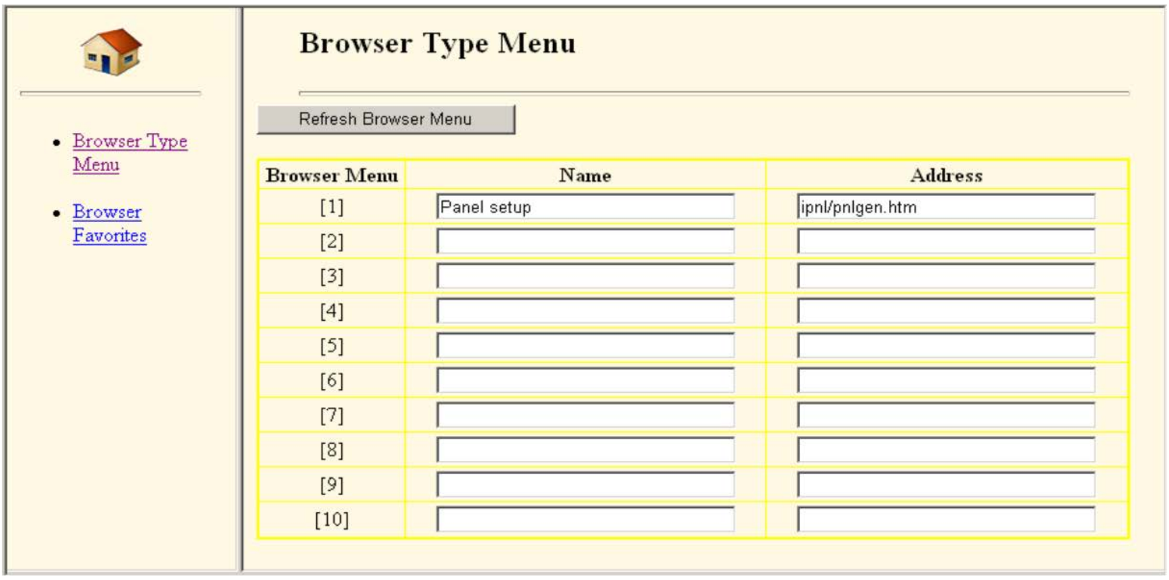
If there are 5000 registers, links like following figure is displayed.



“Refresh Numeric Registers” button updates only registers currently displayed.

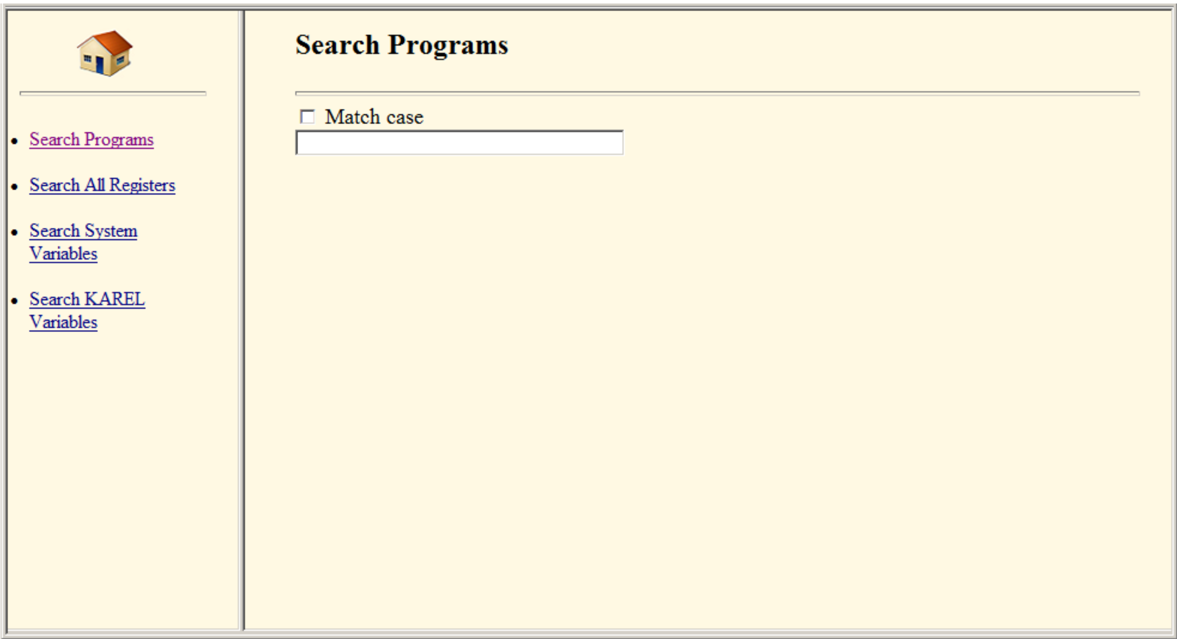
If there are more than 1000 position registers, they are displayed in the same way. The maximum number of position registers displayed in a page is 1000. For more information on Expanded registers function (R830), please refer to chapter of “Expanded registers function” of optional function OPERATOR’S MANUAL (B-83284EN-2).

“Browser Tool” will split the screen as shown in following figure.



Browser Type Menu contains the menu entries of F1 [TYPE] menu of the BROWSER menu. Browser Favorites allow you to add links to the FAVORITE page.

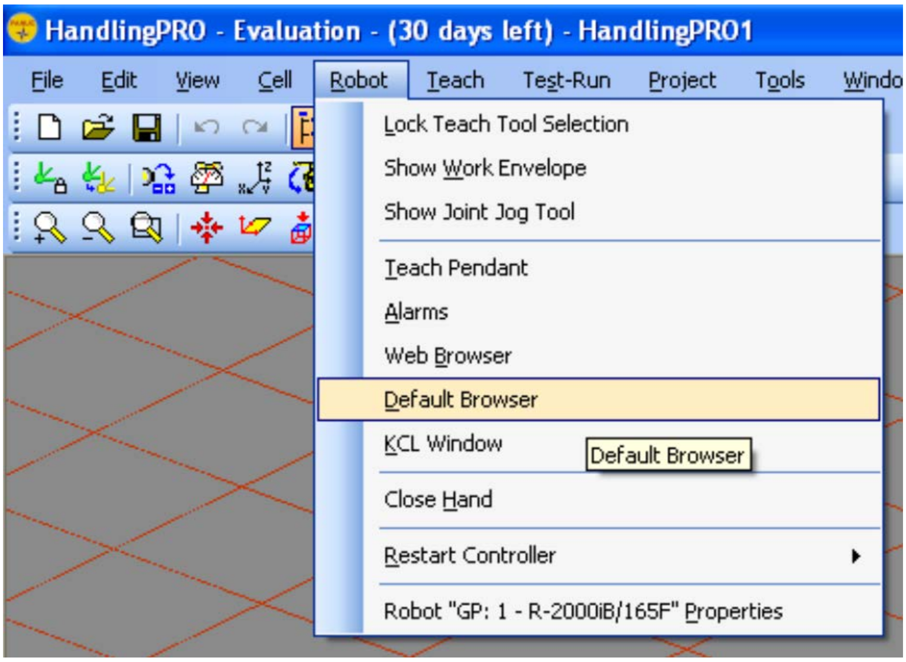
“Search Tool” will split the screen as shown in following figure.



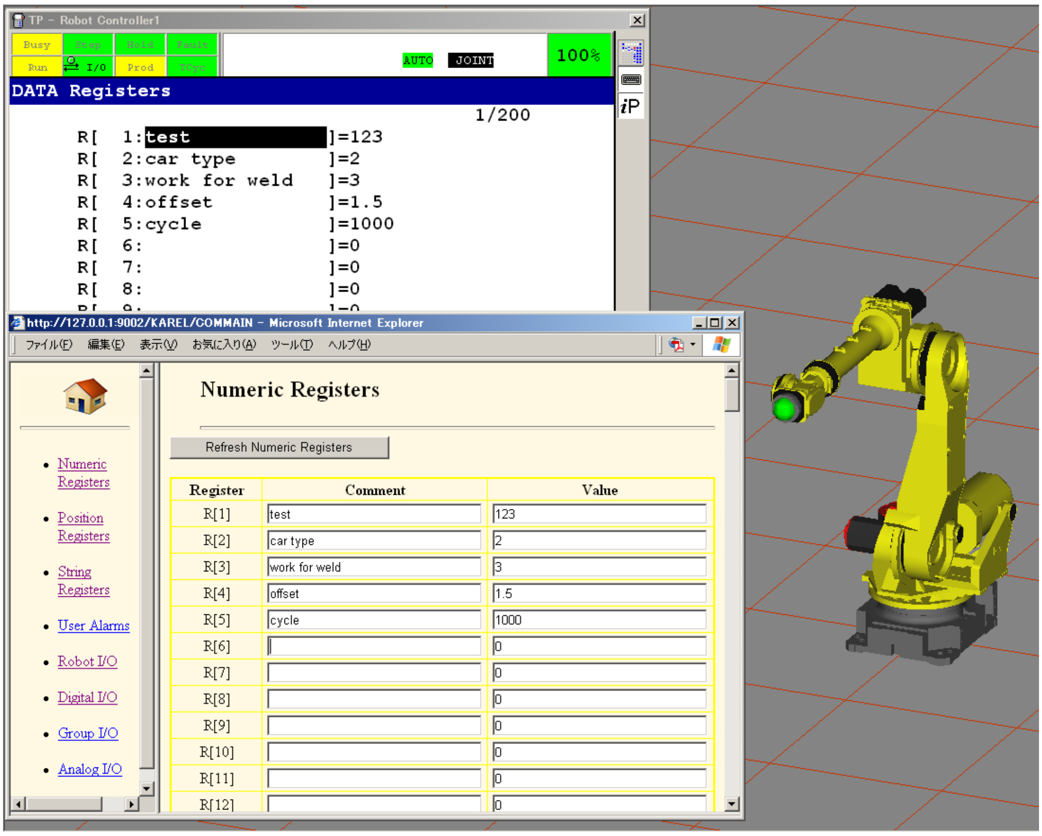
Search Programs will search for text in all programs. It generates a new version of MD:*.LS files and searches within those files. Links to the .LS file will be displayed for all matches.
Search All Registers will search for text in all numeric, position, and string registers, including the comments. It generates a new version of MD:*REG*.VA and searches within those files. Links to the .VA files will be displayed for all matches.
Search System Variables will search for text in all system variables, including the non-saved variables.
It generates a new version of MD:SY*.VA and searches within those files. Links to the .VA files will be displayed for all matches.

Search KAREL Variables will search for text in all KAREL variables. It generates a new version of MD:*.VA, excluding registers and system variables, and searches within those files. Links to the .VA files will be displayed for all matches.
The search is not case sensitive, but you can match the case if desired. The search for match case is faster.

To display ROBOT TOOLS of virtual robot controller by ROBOGUIDE, select “Robot” of menu bar. Then select “Default Browser” from displayed menu.



You can display pages of ROBOT TOOLS like following figure.



9.18 GROUP MASK EXCHANGE

Group Mask Exchange is used to copy a teach pendant program into a new program with a different group mask than an original program. The position data from the original teach pendant program will be copied into another group in the new teach pendant program. You can copy multiple group programs from a single robot or dual robots.

```
Group Mask Exchange
    Program Selection                  2/10
         Orig Prog(*,*,*,*,*,*,*,*)
      1
         New Prog (*,*,*,*,*,*,*,*)
      2
    Group Selection          New Prog Grop
      3 COPY Orig Grp[1] to New Grp[0]
      4 COPY Orig Grp[2] to New Grp[0]
      5 COPY Orig Grp[3] to New Grp[0]
      6 COPY Orig Grp[4] to New Grp[0]
      7 COPY Orig Grp[5] to New Grp[0]
      8 COPY Orig Grp[6] to New Grp[0]

   [ TYPE ]   EXEC                        >
```

NOTE
This feature does not validate the logic of the output program, its positional data, or the program header (except group mask information). The resulting program might also require some position touchup.

Table 9.18 Setting items in group mask exchange screen

Item	Description
Original Program	Select the name of the original program from which to copy the group mask.
New Program	Select the name of the new program to which the group mask will be copied.
Copy Orig Grp[x] to New Grp[x]	Select a motion group in the original program to copy to the new program motion groups where x identifies the group number.

Procedure 9-31 Copying a Program Using Group Mask Exchange

Condition

- The Program on which the exchange is to be performed exists.

```
PROGRAM
                                          1/6
        1: J  P[1] 100% FINE
        2: J  P[2] 70% CNT50
        3: L  P[3] 1000cm/min CNT30
        4: L  P[4] 500mm/sec FINE
        5: J  P[1] 100% FINE
     [End]

   POINT                          TOUCHUP  >
```

Step

1 Press [MENU] key.
2 Select UTILITIES.
3 Press F1, [TYPE].
4 Select Group Exchg and press the [ENTER] key. The following screen will be displayed.

```
Group Mask Exchange
    Program Selection                  2/10
         Orig Prog(*,*,*,*,*,*,*,*)
      1
         New Prog (*,*,*,*,*,*,*,*)
      2
    Group Selection           New Prog Grop
      3 COPY Orig Grp[1] to New Grp[0]
      4 COPY Orig Grp[2] to New Grp[0]
      5 COPY Orig Grp[3] to New Grp[0]
      6 COPY Orig Grp[4] to New Grp[0]
      7 COPY Orig Grp[5] to New Grp[0]
      8 COPY Orig Grp[6] to New Grp[0]

   [ TYPE ]   EXEC                              >
```

> **NOTE**
> If a program is currently selected, it is used as the default for the original and new program displayed in lines 1 and 2 above.

5 To select a different original program, press F4, [CHOICE].

```
Group Mask Exchange
    Program Selection                  1/10
      GETDATA                  ,*,*)
      REQMENU
      SENDDATA                 ,*,*)
      SENDEVENT
      SENDSYSV                New Prog Grop
      TEST1                   v Grp[0]
      TEST2                   v Grp[0]
                              v Grp[0]
                              v Grp[0]
                              v Grp[0]
                              v Grp[0]

   [ TYPE ]   EXEC          [CHOICE]            >
```

6 To rename the new program, move the cursor to line 2 and press the [ENTER] key. You can then choose a method to rename the program and type the new program name. You must then identify the new group for the copy to take place.

```
Group Mask Exchange
    Program Selection                  2/10
         Orig Prog(1,1,*,*,*,*,*,*)
      1 TEST1
         New Prog (*,*,*,*,*,*,*,*)
      2
    Group Selection          New Prog Grop
      3 COPY Orig Grp[1] to New Grp[0]
      4 COPY Orig Grp[2] to New Grp[0]
      5 COPY Orig Grp[3] to New Grp[ Alpha input 1
      6 COPY Orig Grp[4] to New Grp[  Words
      7 COPY Orig Grp[5] to New Grp[  Upper Case
      8 COPY Orig Grp[6] to New Grp[  Lower Case
                                      Options
   PRG    MAIN    SUB    TEST
```

7 To select a motion group in the original program to copy to the new program motion groups:
   a Move the cursor to the first New group [#] to copy.
   b Type a group number, and press the [ENTER] key.

```
Group Mask Exchange
    Program Selection                  3/10
         Orig Prog(1,1,*,*,*,*,*,*)
      1 TEST1
         New Prog (*,*,*,*,*,*,*,*)
      2 TEST2
    Group Selection          New Prog Grop
      3 COPY Orig Grp[1] to New Grp[2]
      4 COPY Orig Grp[2] to New Grp[0]
      5 COPY Orig Grp[3] to New Grp[0]
      6 COPY Orig Grp[4] to New Grp[0]
      7 COPY Orig Grp[5] to New Grp[0]
      8 COPY Orig Grp[6] to New Grp[0]

  [ TYPE ]   EXEC                        >
```

NOTE
If any of the entries for New Group [#] are left at 0, this group will be deleted from the new program.

8 When you have finished identifying the groups to be copied, press F2, EXEC.

NOTE
If the new program exists when you press F2, EXEC, you must confirm the overwrite operation before the execute takes place.

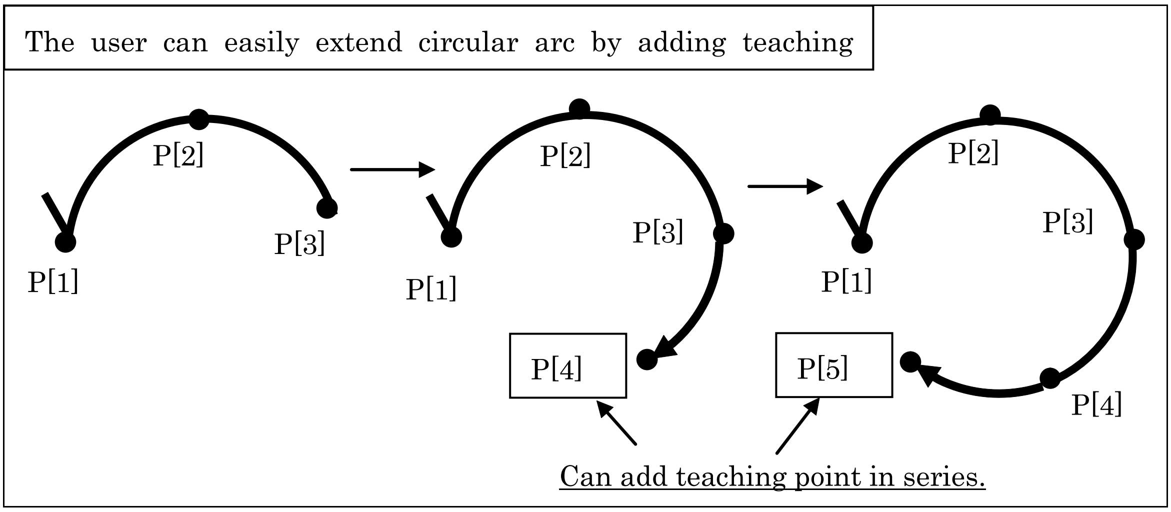
9 To clear a single entry, press NEXT, > and then press F1, CLEAR.
10 To clear all entries, press NEXT, > and then press SHIFT and F1, CLEAR.

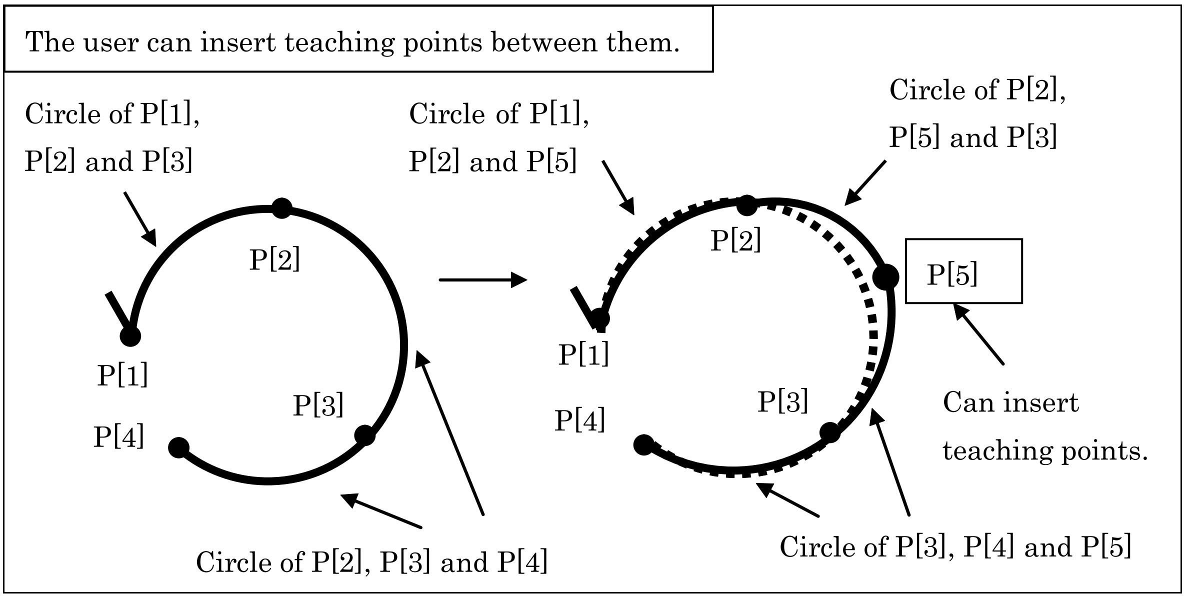
NOTE
If lines 1 and 2 are changed, the New Grp column in lines 3 through 7 are cleared.

9.19 CIRCLE ARC MOTION INSTRUCTION

In the circular motion instruction, the user must record two positions for one circular motion instruction. In the circle arc motion instruction, every instruction has one position, and the circular motion is performed by more than three circle arc motion instructions.
The circle arc motion has the following features.

- Easy to add and delete teaching point on circular arc.
- Speed and CNT can be specified for every position.
- Logical instructions can be written between every position.



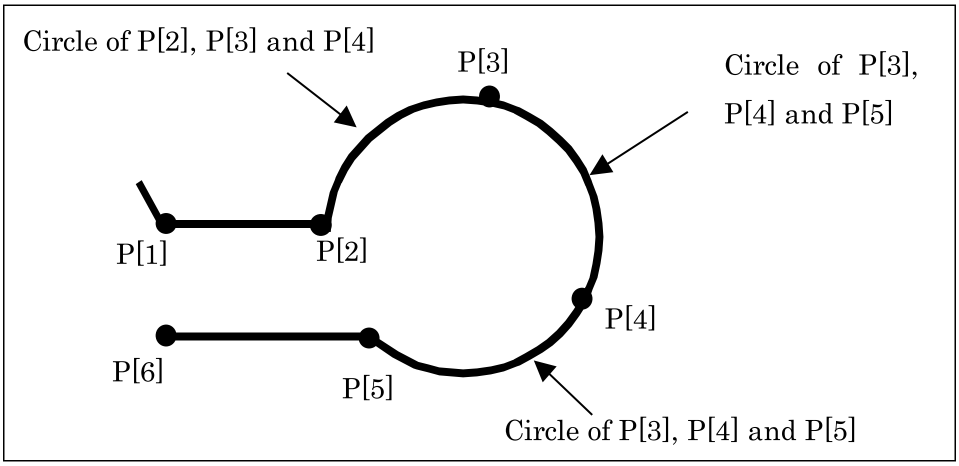


9.19.1 Way of Teaching

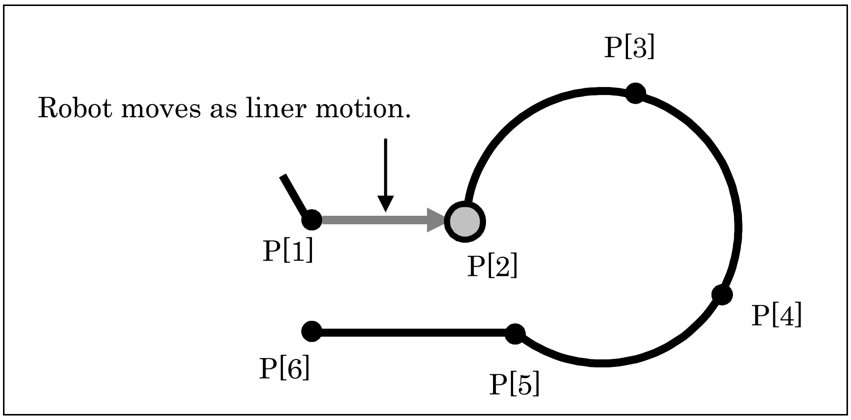
The way of teaching is same as liner motion instruction without select the circle arc motion as a motion type.

9.19.2 Normal Motion

```
Program
                                              1/7
      1: J  P[1] 100% FINE
      2: A  P[2] 200mm/sec FINE
      3: A  P[3] 200mm/sec CNT100
      4: A  P[4] 200mm/sec CNT100
      5: A  P[5] 200mm/sec FINE
      6: L  P[6] 200mm/sec FINE
   [End]

  POINT                          TOUCHUP   >
```

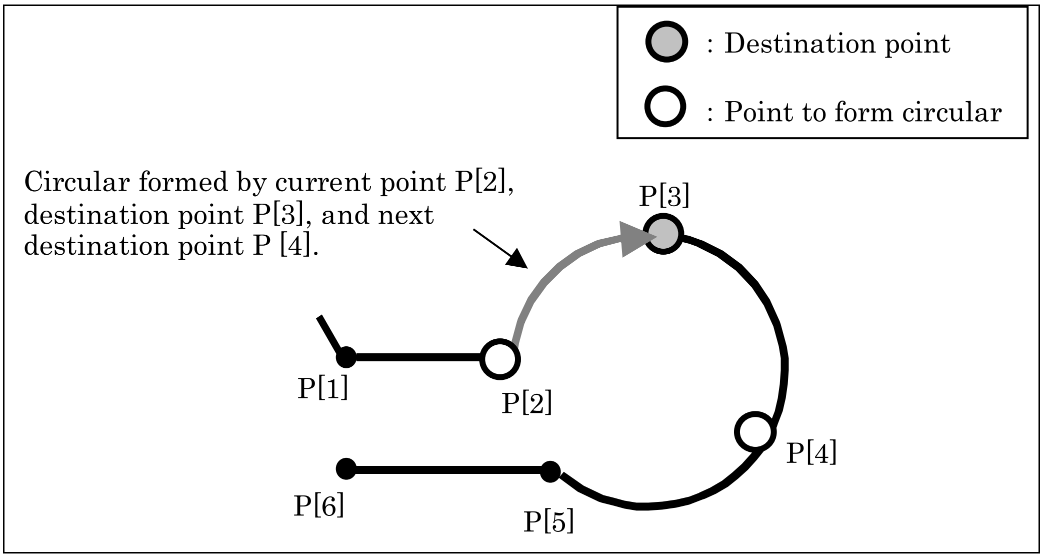


Different from normal circular motion instruction, the motion instruction to start point of circular arc must be the circle arc motion instruction.
In the first circle arc motion instruction, robot moves as liner motion.
<u>First circle arc motion</u>



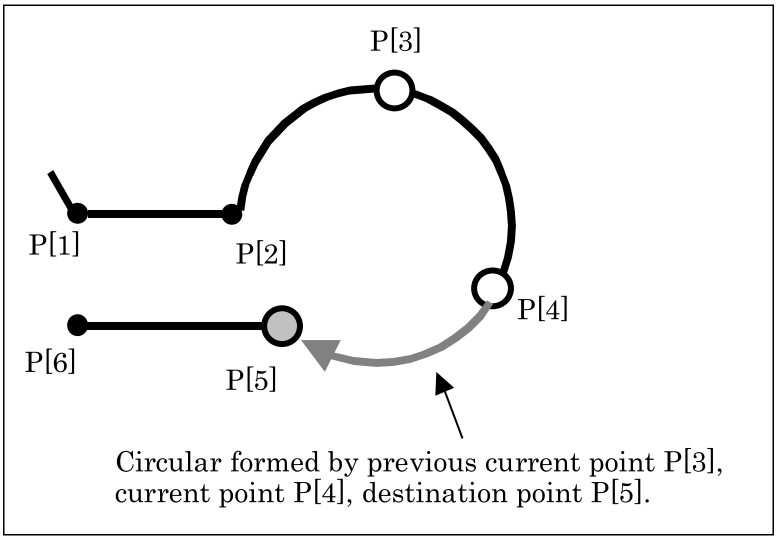
In the second and after the circle arc motion instruction, robot moves on the circle that is calculated by 3 positions. Current position, destination point of this circle arc motion instruction and that of the next circle arc motion instruction.

Halfway circle arc motion



When the next motion instruction is not the circle arc motion instruction, the last circle arc motion instruction is regarded as the end position of the circle. In this case, robot moves on the circle that is calculated by 3 positions, destination position of the previous motion instruction, current position, and destination position of this motion instruction.

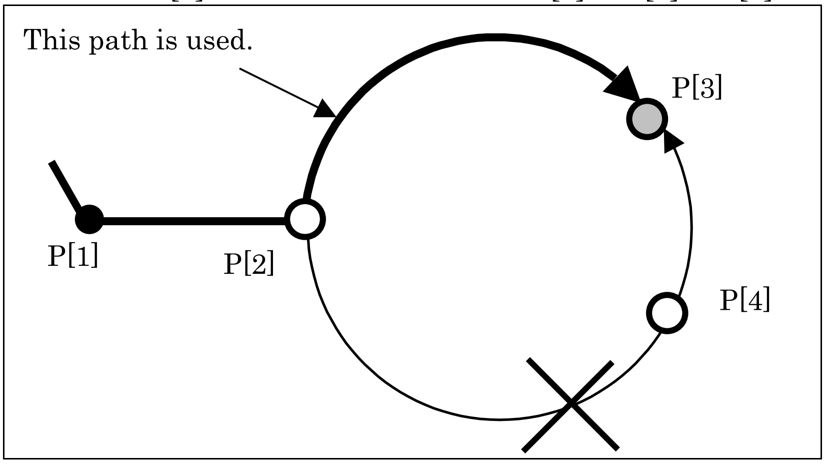
Last circle arc motion



9.19.3 Direction of the Circular Motion

```
Program
                                         3/5
     1: J  P[1] 100% FINE
     2: A  P[2] 200mm/sec FINE
     3: A  P[3] 200mm/sec CNT100
     4: A  P[4] 200mm/sec CNT100
   [End]

   | POINT |  |  |  | TOUCHUP | > |
```

When the 3rd line of above program is executed, robot moves to P[3] on the circle that is calculated by P[2], P[3] and P[4]. Robot moves to P[3] on the direction in order of P[2] → P[3] → P[4].



9.19.4 Cases Circular Path Cannot be Planned

9.19.4.1 Shortage of the circle arc motion instruction

You must teach successive three or more circle arc motion instructions.
If you don’t, it is displayed as “INTP-609 ARC:A needs 3 points ”.

Ex.1) The circle arc motion instruction is isolated. It is displayed as ” INTP-609 ARC:A needs 3 points (1) “.

```
Program
                                                 2/4
     1: J  P[1] 100% FINE
     2: A  P[2] 200mm/sec FINE
     3: L  P[3] 200mm/sec CNT100
   [End]

   POINT                               TOUCHUP  >
```

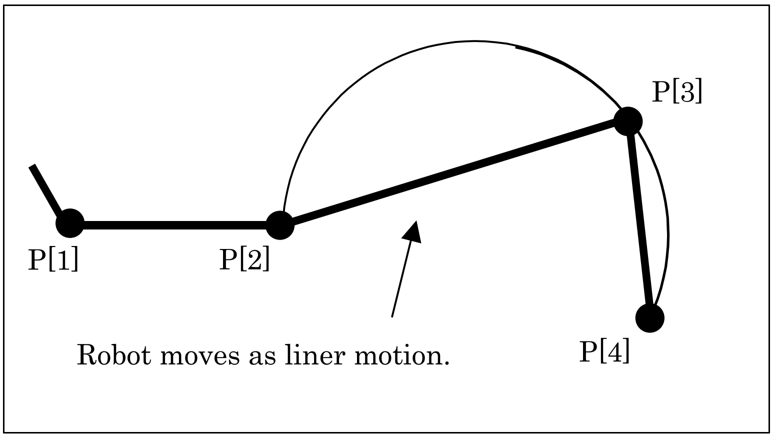
Ex.2) There are only two successive circle arc motion instructions. It is displayed as “INTP-609 ARC:A needs 3 points (2) “

```
Program
                                                 3/5
     1: J  P[1] 100% FINE
     2: A  P[2] 200mm/sec FINE
     3: A  P[3] 200mm/sec CNT100
     4: L  P[4] 200mm/sec CNT100
   [End]

   POINT                               TOUCHUP  >
```

9.19.4.2 Teaching of same position

When it is impossible to form circular by three points, robot moves as liner motion.
When same point was written continuously, robot moves as liner motion.

```
Program
                                                      1/7
     1: J  P[1] 100% FINE
     2: A  P[2] 200mm/sec FINE
     3: A  P[2] 200mm/sec CNT100
     4: A  P[3] 200mm/sec CNT100
     5: A  P[3] 200mm/sec FINE
     6: L  P[4] 200mm/sec FINE
    [End]

   POINT                            TOUCHUP   >
```

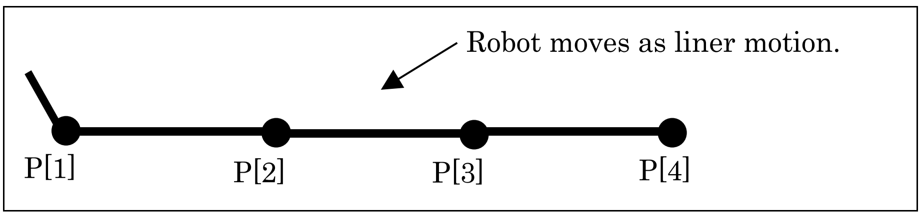


Motion of 3rd line: Current point and destination point are same, so robot doesn't move.
Motion of 4th line: Destination point and next destination point are same, so robot moves as liner motion.
Motion of 5th line: Current point and destination point are same, so robot doesn't move.

9.19.4.3 The case the three points are in line

In this case, robot does not move as circular motion.

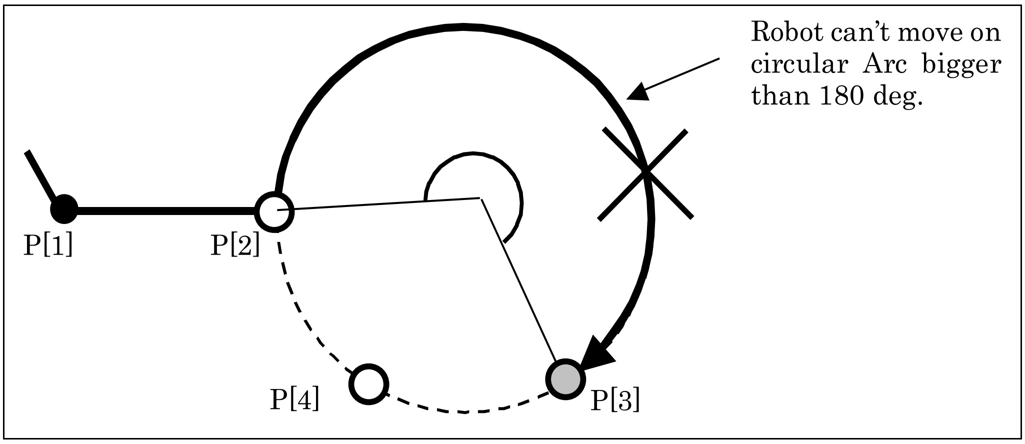
```
Program
                                                      1/5
     1: J  P[1] 100% FINE
     2: A  P[2] 200mm/sec FINE
     3: A  P[3] 200mm/sec CNT100
     4: A  P[4] 200mm/sec FINE
    [End]

   POINT                            TOUCHUP   >
```



9.19.4.4 The case circular arc bigger than 180 deg

In the circle arc motion instruction, robot can't move on circular Arc bigger than 180 deg in a motion.

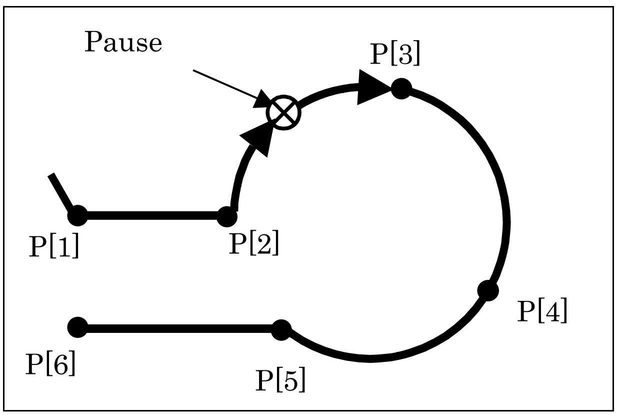
```
Program
                                              1/5
      1: J  P[1] 100% FINE
      2: A  P[2] 200mm/sec FINE
      3: A  P[3] 200mm/sec CNT100
      4: A  P[4] 200mm/sec CNT100
   [End]

   | POINT |     |     |     | TOUCHUP | > |
```



9.19.5 Resume After Pause

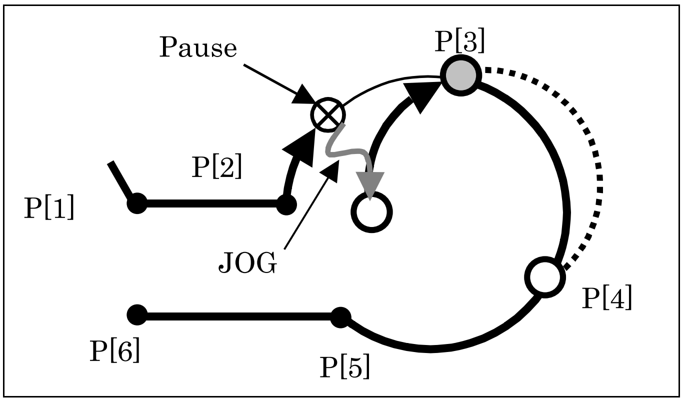
In case robot motion is paused and resumed without change, resumed path is just as usual.

```
Program
                                              3/7
      1: J  P[1] 100% FINE
      2: A  P[2] 200mm/sec FINE
      3: A  P[3] 200mm/sec CNT100
      4: A  P[4] 200mm/sec CNT100
      5: A  P[5] 200mm/sec FINE
      6: L  P[6] 200mm/sec FINE
   [End]

   | POINT |     |     |     | TOUCHUP | > |
```



9.19.5.1 Resume after JOG

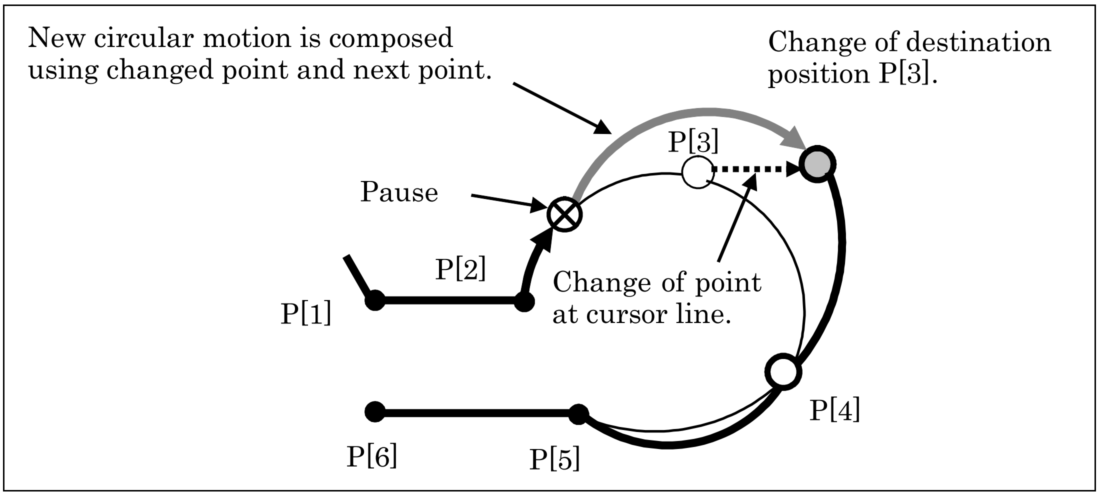
Suppose that a program is paused during the circle arc motion. Then robot is jogged and the program is resumed. In this case, if original path resume is enabled, robot returns to the paused position, and continue the motion. (If original path resume is disabled, robot moves from current position to destination position on circular path. Then robot moves along programmed path.)
For example, program is paused during motion at 3rd line. Then the robot is jogged and the program is resumed. At first, robot moves to P[3] as circular motion made by current position P[3] and P[4]. Then it moves to P[4] on the circular arc formed by P[3], P[4] and P[5]. The path to P[3] and the path to P[4] are blended because motion instruction to P[3] uses CNT100.



9.19.6 Resume After Pause and Modification of Programs

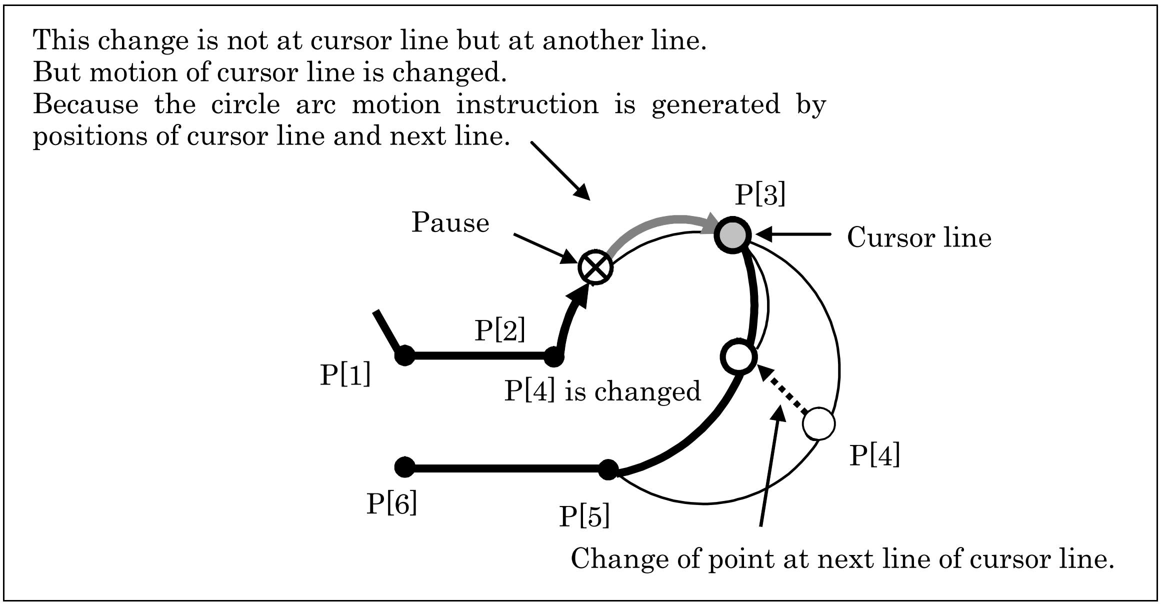
9.19.6.1 Change of destination point

In case that the program is resumed after the program is paused during the circle arc motion and the destination point is modified, the resume motion is circular motion and its path is calculated using changed destination point.
For example, suppose program is paused during motion to P[3]. Then P[3] is modified by direct numerical input and program is resumed. Robot moves as circular motion that is calculated by current position, destination position P[3] and next position P[4].



9.19.6.2 Change of next destination point

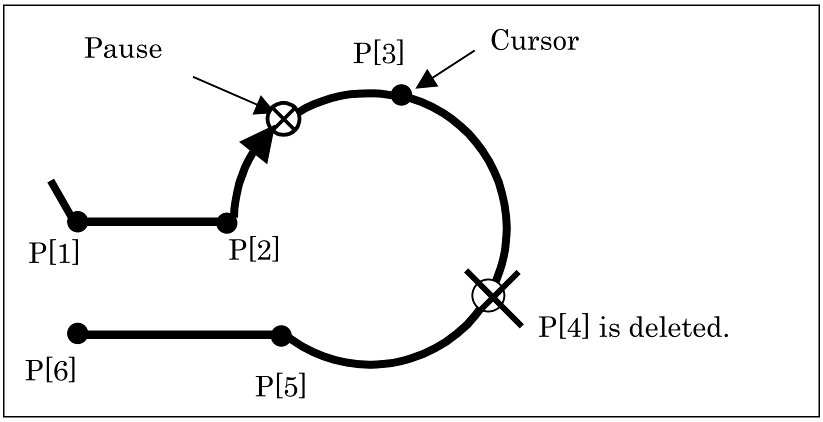
If P[4] is changed in example above, changed P[4] is used for calculation of circular path.



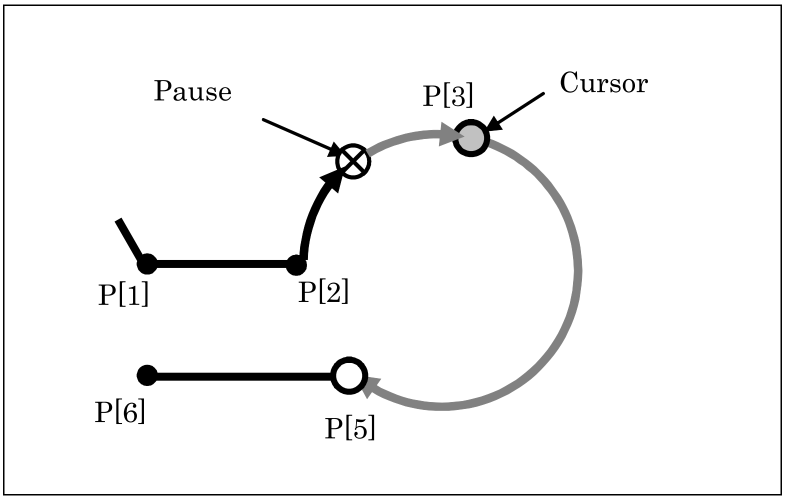
9.19.6.3 Deletion of next circle arc motion instruction and resume

If program is paused on the circle arc motion and next motion destination point is deleted, it means that point used to calculate circular path is lost. Even after deletion, if the next destination point is the circle arc motion instruction, resumed motion is circular arc. It is calculated by current position, destination point and next destination point.

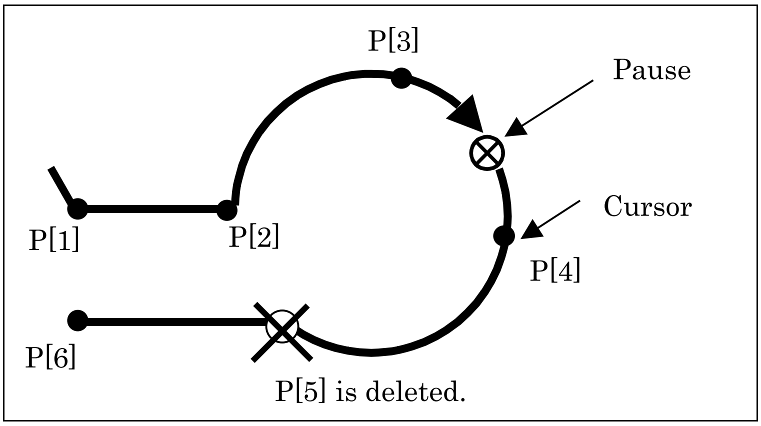
Following figure is an example. Program is paused during motion from P[2] to P[3], and P[4] is deleted.



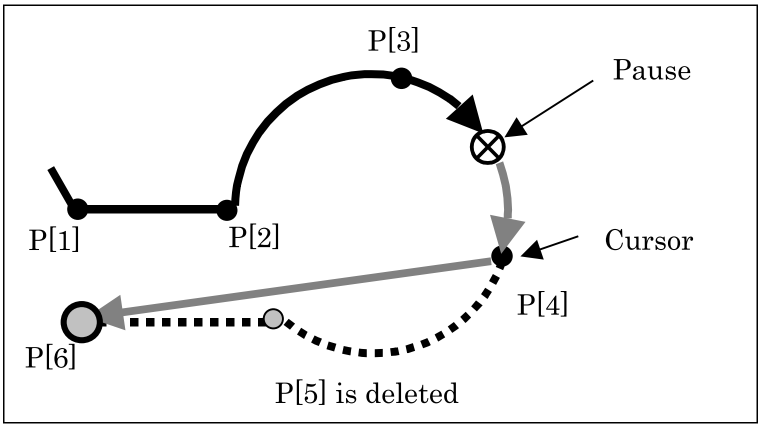
To P[3], robot moves as circular motion. The arc is formed by current position, P[3] and P[5].
To P[5], robot moves as circular motion that is calculated by P[2], P[3] and P[5].



If there is no more motion instruction or next motion instruction isn't the circle arc motion, paused motion instruction is the last circle arc motion. In this case, circular motion is calculated by current position, and destination position.
Suppose that program is paused during the motion from P[3] to P[4]. Then motion instruction to P[5] is deleted.



Because P[5] is deleted, motion to P[4] is the last circle arc motion. After resume robot moves to P[4] as the circle arc motion that is calculated by P[3], current position and P[4].
After that robot moves to P[6] as liner motion.



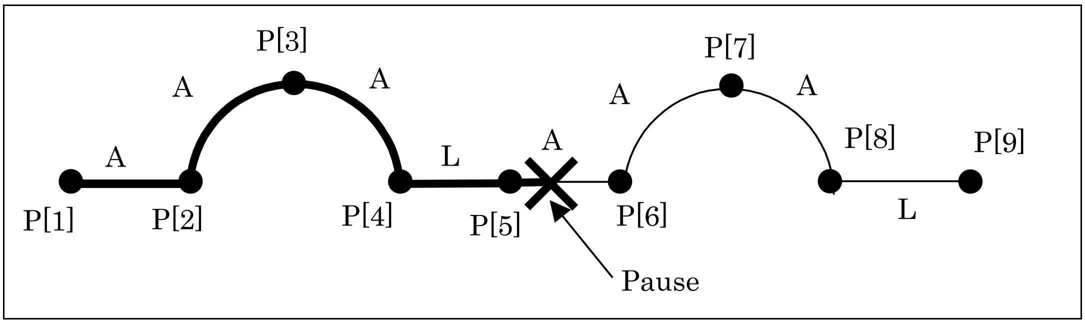
9.19.6.4 Current instruction becomes the first circle arc motion after modification

Suppose that program is paused on the first circle arc motion and modified. Even if program is changed and the paused motion instruction isn't the first one any more, robot moves as liner motion.
On contrary, even if the circle arc motion instruction is changed to the first one after modification of program, the motion is resumed as the circle arc motion.
Change is not reflected.

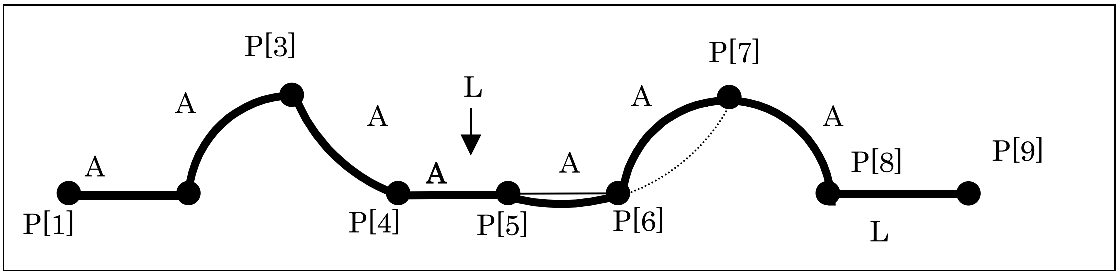
```
Program
                                   1/10
      1: J  P[1] 100% FINE
      2: A  P[2] 200mm/sec FINE
      3: A  P[3] 200mm/sec CNT100
      4: A  P[4] 200mm/sec FINE
      5: L  P[5] 200mm/sec CNT100
      6: A  P[6] 200mm/sec FINE
      7: A  P[7] 200mm/sec CNT100
      8: A  P[8] 200mm/sec FINE
      9: L  P[9] 200mm/sec FINE
     [End]
   POINT                    TOUCHUP  >
```

Ex1)

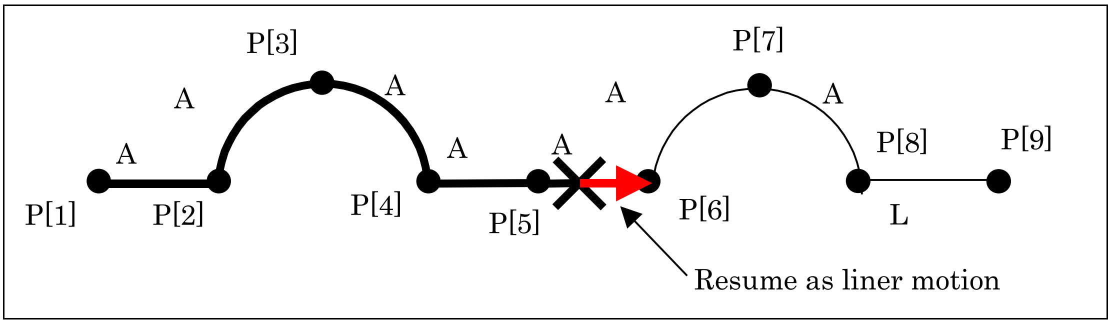
Program is paused during the motion to P[6] and change the previous instruction to P[5] from the liner motion to the circle arc motion.



Motion to P[6] was the first circle arc motion, so it was liner motion. But after modification programmed path is as follows.



Even in this situation, if motion to P[6] is resumed, robot moves to P[6] as liner motion.

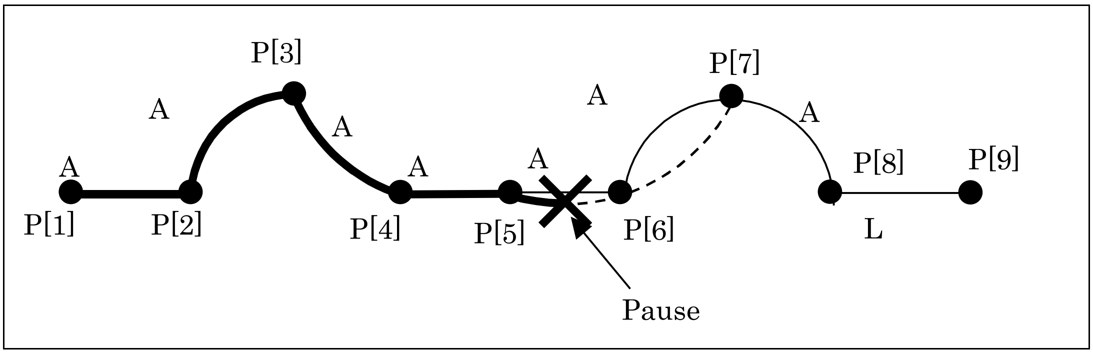


But if program is resumed by backward execution, robot moves to P[5] as circular motion. Resumed path is calculated by P[7], current position and P[5].

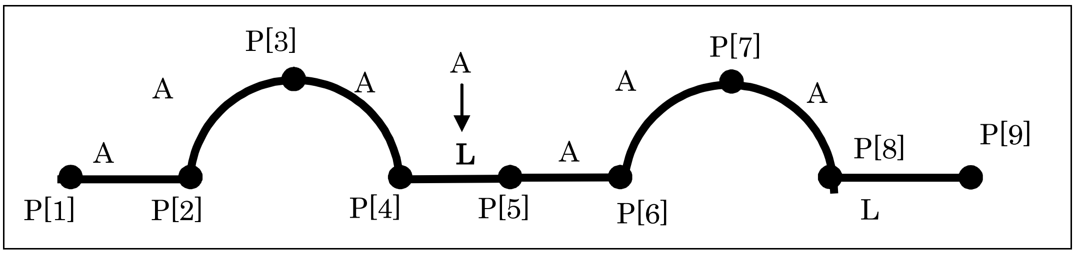
Ex.2)

Changes contrary to Ex.1.

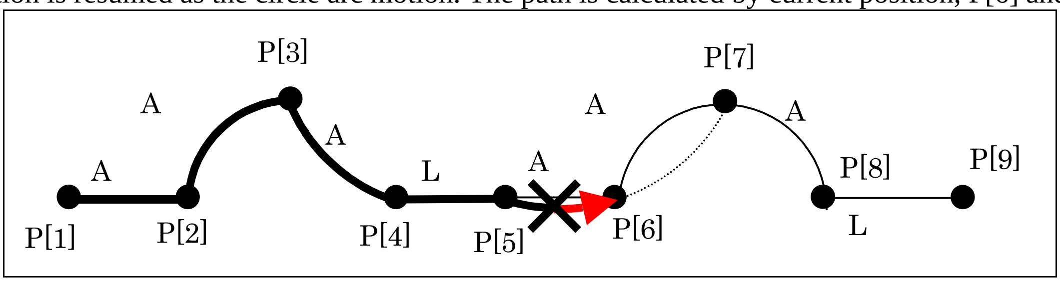
Program is paused during the motion to P[6]. The instruction to P[5] is changed to liner motion.



Following figure is programmed path of modified program. The motion instruction to P[6] is the first instruction now.



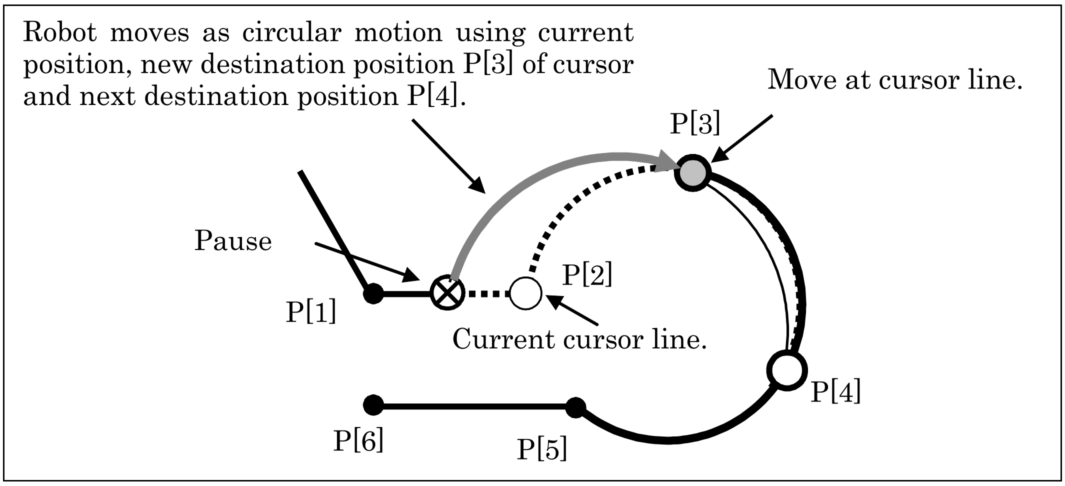
But motion is resumed as the circle arc motion. The path is calculated by current position, P[6] and P[7].



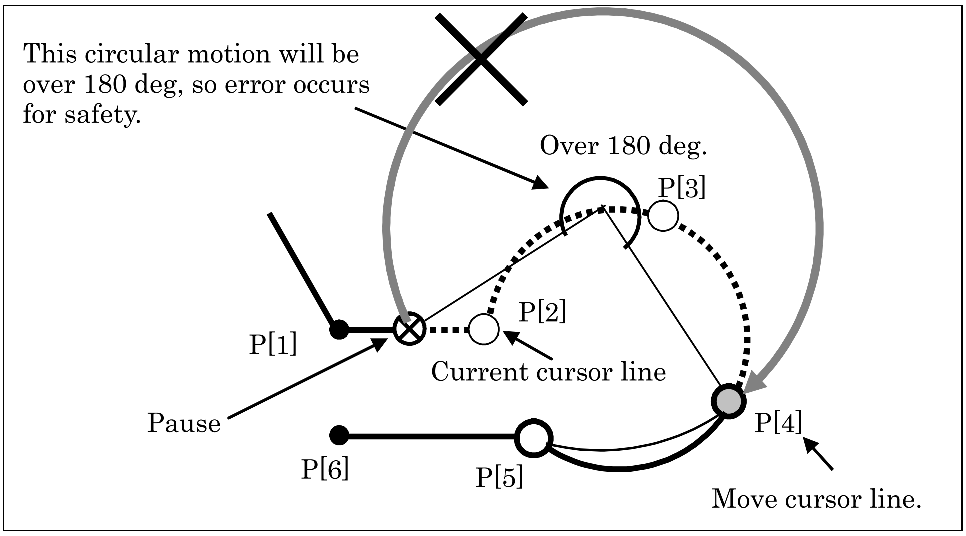
If program is resumed by backward execution, robot moves to P[5] as liner motion.

9.19.7 Pause and Resume from Another Circle Arc Motion Instruction

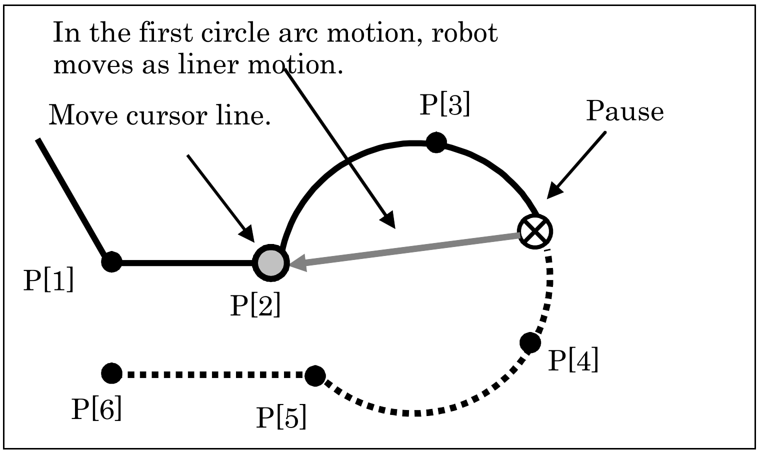
The circle arc motion is resumed same as other motion types.
In following example, program is paused during motion to P[2] and resumed from motion instruction to P[3].
Robot moves as circular motion in order to current position, P[3] and P[4].



In following example, program is paused during motion to P[2] and resumed from motion instruction to P[4].
This path will be as gray line in the following figure. But this circular motion will be over 180 deg, so error "MOTN-098 Circle angle too large" occurs and the program isn't executed.



Next example is resume from motion instruction to P[2] after hold during motion instruction to P[4]. Motion instruction to P[2] is the first circle arc motion. So robot moves as liner motion.

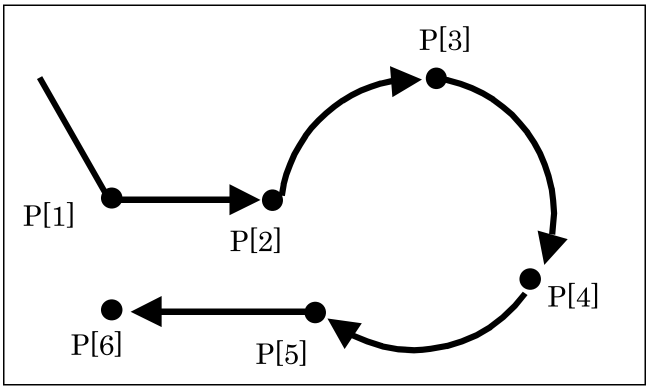


9.19.8 Start of Program from the Circle Arc Motion

This case is same as “ 9.18.7 Pause and resume from another circle arc motion instruction “.

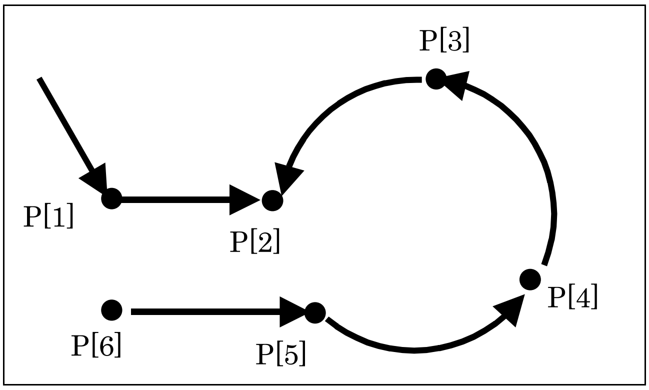
9.19.9 Single Step Execution

If the circle arc motion instruction is executed by single step, program is paused on every taught point like another motion instruction.



9.19.10 Backward Execution

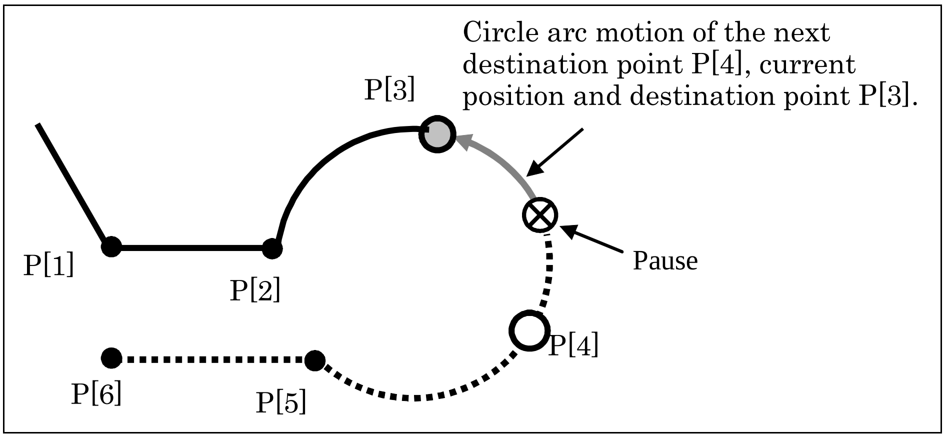
If the circle arc motion instruction is executed by backward, robot moves as backward on the same path as forward execution, and the program is paused on every taught point.



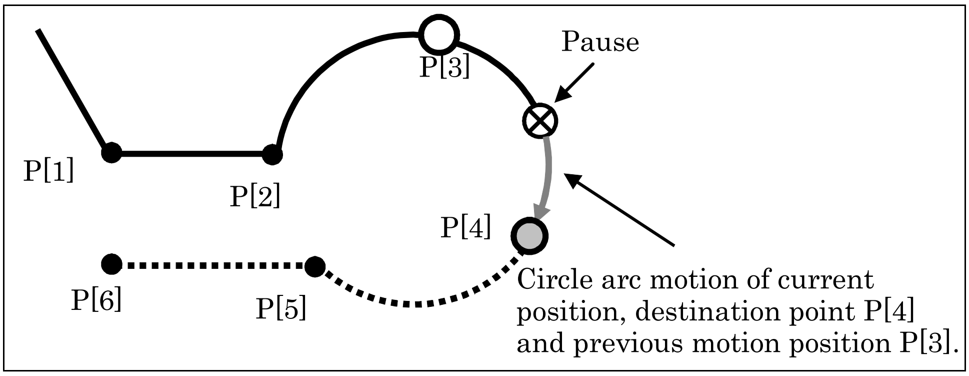
9.19.11 Backward Execution after Abort

In case of the last circle arc motion, robot moves as liner motion.
In other cases, robot moves as circular motion. Positions used to calculate the circular motion are as follows.
The circle arc motion instruction just before the last circle arc motion instruction:
Current position, destination position and the position of the previous circle arc motion
Other case:
Destination position of the next circle arc motion, current position and destination point.

The following diagram shows an example. Program is aborted during motion to P[4]. The program is resumed by BWD, toward P[3].



If program is started by BWD from motion to P[4] from the same position, the line is the 2nd last circle arc.



9.19.12 Backward Resume from Different Line

Same as “ 9.18.11 Backward execution after abort”.

9.19.13 Logic Instructions between the Circle Arc Motion

Logic instructions can be taught between the circle arc motion instructions.
The circle arc motion instruction uses positions of successive circle arc motion instructions to calculate circular path. Logic instructions that change the order of execution such as JMP, LBL cannot be executed between the circle arc motions. Logic instructions that effect on position data such as position register instruction neither.

For example, execution of the following programs causes error at line 4. P[5] is used to calculate the circle arc motion to P[4], so error occurs before execution of 4th instruction to prevent unexpected motion.

```
Program
                                              1/8
      1: J  P[1] 100% FINE
      2: A  P[2] 200mm/sec FINE
      3: A  P[3] 200mm/sec CNT100
      4: A  P[4] 200mm/sec CNT100
      5: JMP LBL[1]
      6: A  P[5] 200mm/sec FINE
      7: L  P[6] 200mm/sec FINE
   [End]

   POINT                          TOUCHUP  >
```

Available logic instructions between the circle arc motion instructions

Instructions	Examples	Notes
Register instruction	R[1]=1	
I/O instruction	DO[1]=ON DO[1]=PULSE	DO, RO, UO, SO, WO, GO, AO and F are available.
Wait	WAIT 2.0sec WAIT DI[1]=ON	WAIT /TIMEOUT/ LBL causes error.
Call	CALL (Prg)	If called program has motion instruction or unavailable logic instruction between the circle arc motion instructions, error occurs.
Macro	HAND1 OPEN	If macro program has motion instruction or unavailable logic instruction between the circle arc motion instructions, error occurs.
If / Call	IF DI[1]=ON CALL (Prg)	IF / JMP / LBL causes error.
Select / Call	SELECT R[1]=1 CALL (Prg1) =1 CALL (Prg2) OTHERS CALL(Prg3)	SELECT / JMP / LBL causes error.
RSR	RSR[1]	
User alarm	UARM[1]	
Override	OVERRIDE=50%	
Timer	TIMER[1]=START	
Parameter	$(Parameter)=100	
Remark Multi language remark	! (Remark) -- (Remark)	
TP line remark	// J P[1] 100% FINE	
Message	Message[message statement]	
Positional register lock / unlock	LOCK PREG UNLOCK PREG	
Maximum speed	JOINT_MAX_SPEED[1]=50	
Pause	PAUSE	
Program execution instruction	RUN (program name)	
Torque limit	TORQ_LIMIT 50.0%	
Monitor instruction	MONITER (Prg) MONITER END (Prg)	Contents of the program specified by monitor instruction are not checked.
Condition	CONDITION R[1]=1 CALL(Prg)	Condition instruction isn't checked because it is taught only in program specified by monitor instruction.
InterBus-S	IBS ATTACH	
InterBus PCI	IB attach	
Path switch	PS -10mm,1.0sec DO[1]=ON	
Collision detect	COL DETECT ON	

Instructions	Examples	Notes
Approach inst	Approach_STOP[1] = TMP_DISABLE	
Arc	Arc Start[1]	
ERROR_PROG	ERROR_PROG=Prg	
RESUME_PROG	RESUME_PROG[1:Comment]=Prg	
CLEAR_RESUME_PROG	CLEAR_RESUME_PROG	
RETURN_PATH_DSBL	RETURN_PATH_DSBL	
MAINT_PROG	MAINT_PROG[1:Comment]=Prg	
TC_ONLINE	TC_ONLINE(!DI[1])	
Data monitor	MONITOR[1] MONITOR END	

Instructions unavailable between the circle arc motions but available in called programs

Execution of following instructions between the circle arc motion instructions causes error. But they are available in programs called between the circle arc motion instructions.

Instructions	Examples	Notes
End	END	
Jump	JMP LBL[1]	
Label	LBL[1]	

Instructions unavailable between the circle arc motion instructions

Instructions not given above examples are unavailable between the circle arc motion instructions. Major unavailable instructions are as follows.

Instructions	Examples	Notes
Position register	PREG[1]=LPOS PREG[1,1]=100	
Abort	ABORT	
Offset	OFFSET_CONDITION PR[1]	
Tool offset	TOOL_OFFSET_CONDITION PR[1]	
Vision offset	VOFFSET VR[1]	
User frame User frame number Tool frame Tool frame number	UFRAME[1]=PR[1] UFRAME_NUM=1 UTOOL[1]=PR[1] UTOOL_NUM=1	
Spot tool	SPOT[1]	
Palletizing	PALLETIZING-B_1	
Simultaneous group instruction Independent group instruction	Simultaneous GP Independent GP	

9.19.14 Available Motion Options

Some motion options cannot be used with the circle arc motion instruction. Unavailable motion options aren't displayed in motion option sub-menu.

Available Motion Options

Instructions	Example	Comment
Wrist joint	WJNT	
Acceleration override	ACC 80	
Path	PTH	

Instructions	Example	Comment
Offset	Offset PR[1]	
Tool Offset	Tool_Offset PR[1]	
Remote TCP	RTCP	
Time Before Time After Distance Before	TB 2.00sec CALL PRG TA 0.30sec CALL PRG DB10.0mm CALL PRG	In case there is a logic instruction forbidden between the circle arc motion instructions in called program, error is posted when executed.
Arc Welding	Arc Start[1]	
Weave	Weave Sine[1]	
Coordinated motion	COORD	
Corner Region	CR50	
Process speed	PSPD 50	
Vision offset	VOFFSET	
Continuous Rotation	CTV100	
Minimal rotation	WJNT MROT	

Major unavailable Motion Options

Motion Options not in above table cannot be added to the circle arc motion instruction. Major unavailable Motion Options are as below.

Instructions	Example	Comment
Incremental	INC	
Skip	Skip LBL[1]	
High Speed Skip	Skip PR[1] LBL[1]	
Spot Welding	SPOT[1]	
Dispense start/end	SS[1]/SE	SS[1] and SE cannot be added to A motion. However the circle arc motion is available between SS and SE.
Touch Sensing	Search [X]	

9.19.15 Change of Position Data during Execution of the Circle Arc Motion Instruction

In program with the circle arc motion instructions, one position data is accessed by multiple motion instructions. So in case the position data is changed while the data is being used for calculation of circular path it causes contradiction between calculated paths. Robot can move along unexpected path. To avoid this, the circle arc motion instruction can detect inconsistent position at execution. If detected, alarm is posted.

```
Program
                                          1/8
     1: J  P[1] 100% FINE
     2: A  P[2] 200mm/sec CNT100
     3: A  P[3] 200mm/sec CNT100
     4: A  PR[1] 200mm/sec CNT100
     5: A  P[5] 200mm/sec CNT100
     6: A  P[6] 200mm/sec CNT100
     7: L  P[7] 200mm/sec FINE
    [End]

   POINT                        TOUCHUP  >
```

Line 4 uses PR[1].
When line 3 is executed, circular path is calculated by 3 points P[2], P[3] and PR[1]. So position data of PR[1] is used.
When line 6 of this program is executed, circular route is computed by 3 points PR[1], P[5] and P[6]. So position data of PR[1] is used again.
Therefore, if position in PR[1] is changed during execution from line 3 to line 6, robot can move along unexpected path. To avoid this, substitution to position register is forbidden between the circle arc motion instructions.
But you can change the value of position register in position register screen. The circle arc motion instruction can detect change of position register. At every execution of instructions, it checks if the value of position register is changed or not.

For example, if the value of the PR[1] is changed in position register screen during execution of line 5, “INTP-627 (program name, line number) ARC: Inconsistent position” is posted when line 6 is executed.

To avoid stopping production by changing position register, you can use “LOCK PREG“/”UNLOCK PREG“ instructions. Teach “LOCK PREG“/”UNLOCK PREG“ before/after the circle arc motion instructions that access position register. Following program is an example. This makes it is impossible to change the value of position register by position resister screen when instructions between “LOCK PREG“ and “UNLOCK PREG“ are executed.

```
Program
                                            1/10
      1: J  P[1] 100% FINE
      2:   LOCK PREG
      3: A  P[2] 200mm/sec CNT100
      4: A  P[3] 200mm/sec CNT100
      5: A  PR[1] 200mm/sec CNT100
      6: A  P[5] 200mm/sec CNT100
      7: A  P[6] 200mm/sec CNT100
      8:   UNLOCK PREG
      9: L  P[7] 200mm/sec FINE
     [End]
   POINT                              TOUCHUP  >
```

Change of User frame or Tool frame also has effect on position data in the circle arc motion instructions. Therefore instructions that change User frame or Tool frame can’t be executed between the circle arc motion instructions.
But you can change User frame or Tool frame by using “SETUP Frames” screen. The circle arc motion instruction can detect the change of User frame or Tool frame. At every execution of instruction, it checks if User and Tool frame is changed or not.

9.19.16 Restrictions

- Backward execution from interrupt program is not supported.
- In case PAL_*[ ] is used as position of the circle arc motion instruction, error is posted when it is executed.
- In case position register is used as the position the circle arc motion instruction and its index is specified indirectly by register, error is posted when the motion is executed.
- In case Offset or Tool_Offset is attached to the circle arc motion instruction and index of PR is indirectly specified, error is posted when the motion is executed.

9.20 *i*RCALIBRATION VISION MASTER RECOVERY

9.20.1 Overview of Vision Master Recovery

Vision Master Recovery is a function to reactivate the robot mastering data, which are also called master counts and mean zero positions of rotational axes, when maintenance operation such as replacement of motors or reducers is done and the data is invalidated. By this function a camera located on the robot arm automatically measures visual target marks fixed near the robot before and after the maintenance operation. It allows users to recover the mastered status of the robot easily and precisely.

9.20.2 Features and Limitations of Vision Master Recovery

- Because the operation of Vision Master Recovery is executed automatically except the camera installation, the usage is very simple and the result does not depend on skill of the operator.
- When the mastering data of multiple axes are invalidated by the maintenance operation, Vision Master Recovery can recover the mastering status of those axes at the same time.
- This function is only available for robot that can move before replacement of the robot parts.

9.20.3 System Configuration for Executing Vision Master Recovery

To execute Vision Master Recovery, the devices as shown in Fig 9.20.3, including the teach pendant, a camera, a lens, a camera cable and a calibration grid are required.

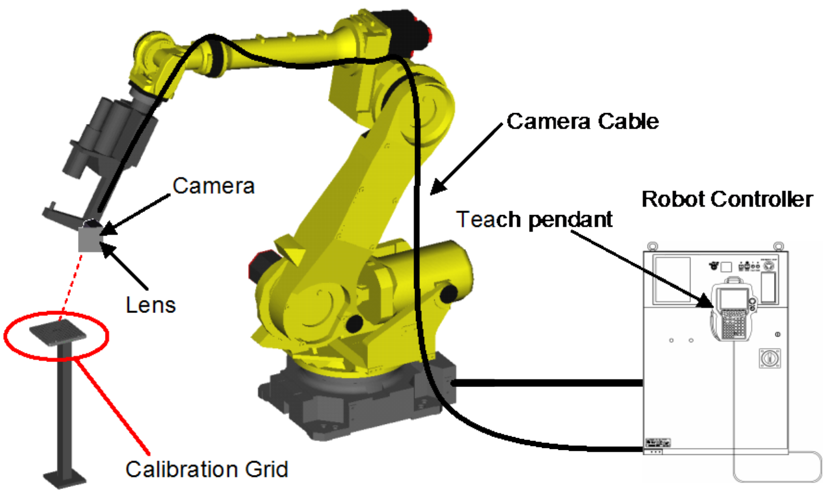


Fig 9.20.3 System configuration for executing vision master recovery

> ⚠ **CAUTION**
> 1 Several types of camera, camera cable, lens, and calibration grid can be provided from FANUC.
> 2 A jig for fixing the camera to end effector of the robot and a jig for locating the calibration grid need to be prepared by customers.
> 3 For details of the operation of Vision Master Recovery, please refer to the "FANUC Robot series R-30*i*B/R-30*i*B Mate CONTROLLER *i*RCalibration OPERATOR'S MANUAL" (B-83724EN).

9.21 OVERVIEW OF KAREL

9.21.1 What is KAREL?

KAREL is the robot language for the robot system architecture. User’s original function can be created after creating KAREL program on PC, loading it to the robot controller and executing it.

The program created with KAREL is able to execute on the robot controller. It is the same of the program of creating on teach pendant (called TP program hereinafter to distinguish KAREL program) in the point that it can be executed on the robot controller. Different point is the purpose of using. TP program is to execute robot motion and application instruction. KAREL program is to build up the robot system.

TP program can be created, edited, and executed on teach pendant, but KAREL program cannot be created and edited on the robot controller. Creating KAREL program on PC and converting it (called translate hereinafter), then execution form program is executed after loading it to the robot controller. TP program can be changed in necessity of usual operating, but KAREL program is created the time of building up the robot system and never to be changed in usual operating.

For more detail, please refer to the “FANUC Robot series R-30*i*A/ R-30*i*A Mate CONTROLLER KAREL Function OPERATOR’S MANUAL” (B-83144EN) and FANUC Robot series KAREL Function OPERATOR’S MANUAL(Reference)” (B-83144EN-1) .

9.21.2 Feature

The following are list of feature of KAREL.

- Various built-in function that allows you to use many function of the robot controller.
- I/O and variable event handling independent of TP program sequence.
- Input/Output by file or key operation.
- Data transfer from/to external by serial port or Ethernet.

KAREL has functions of handling vector and position data and I/O events independent of the robot controller. Furthermore many built-in functions that control the robots or the robot controller are prepared as the standard. Users can create their own system using these functions. For more detail of built-in functions, please refer to the “FANUC Robot series R-30*i*A/ R-30*i*A Mate CONTROLLER KAREL Function OPERATOR’S MANUAL” (B-83144EN).

TP program features controlling the robot motion sequence. On the other hand, KAREL program features control of the other functions except motion control. KAREL can realize the user or system original function not changing the robot controller system software.

9.21.3 Setup before Use of KAREL

Set system variable $KAREL_ENB to 1 in order to use KAREL program. With this setting you can do the following.

- KAREL programs are allowed to display in SELECT screen.
- When CALL or RUN instruction is taught, KAREL programs are allowed to be selected just like TP programs.

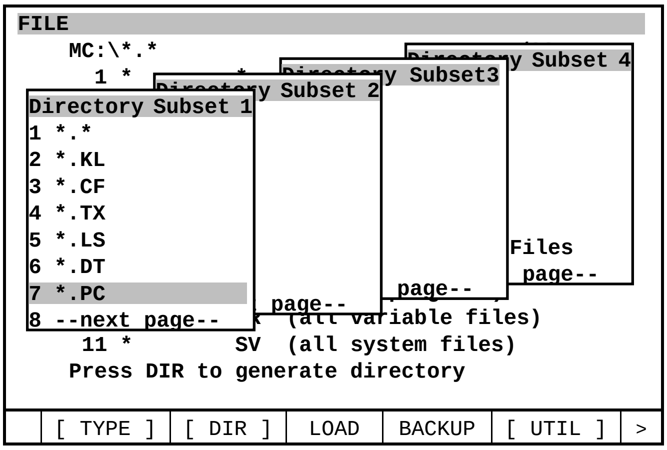
> **NOTE**
> System variable $KAREL_ENB can be 1 by default depending on option software configuration.

9.21.4 How to Load KAREL Program

KAREL program is created by ROBOGUIDE. ROBOGUIDE is off-line programming PC software for FANUC robot. Created KAREL program has file extension, PC. KAREL programs can be loaded to robot controller via file device (memory card, MC: for example). Assuming file device is "MC:", we describe how to load KAREL program.

Step

1 Copy your KAREL program to root directory of memory card.
2 Insert the card to memory card slot of robot controller.
3 Press [MENU] key.
4 Select "7 FILE".
5 If MC: is not selected as file device, press F5, [UTIL] and select Set Device. Select "Mem Card (MC:)" from displayed list.
6 Press F2, [DIR] and select "7 *.PC".



7 Move cursor to KAREL program you created and press F3, LOAD.
8 Confirmation message is displayed. Press F4, YES.

Loaded KAREL programs are displayed in SELECT screen. In SELECT screen, if "PC" is displayed on right side of program name, the program is KAREL program.

9.21.5 How to Run KAREL Program

Loaded KAREL program can be run just as TP program. Following Subsections show examples.

9.21.5.1 Running KAREL program using SELECT screen

Step

1 Press [SELECT] key to display SELECT screen.

```
SELECT
                1014788 bytes free    9/10
     No.  Program name        Comment
      1  -BCKEDT-          [                 ]
      2  GETDATA       MR [Get PC Data       ]
      3  REQMENU       MR [Request PC Menu   ]
      4  SENDDATA      MR [Send PC Data      ]
      5  SENDEVNT      MR [Send PC Event     ]
      6  SENDSYSV      MR [Send PC Sysvar    ]
      7  TEST1         PC [TEST PROGRAM1     ]
      8  TEST2         PC [TEST PROGRAM2     ]
      9  KRLPRG        PC [KAREL PROGRAM     ]
     10  ZPRG001       PC [Program001        ]

 [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR]  >
```

2 Move cursor to KAREL program to be run and press the [ENTER] key. Selected program is displayed in upper part of teach pendant screen.

Selected program

```
KRLPRG  Line 0  ABORTED   JOINT          10%
SELECT
                1014788 bytes free    9/10
     No.  Program name        Comment
      1  -BCKEDT-          [                 ]
      2  GETDATA       MR [Get PC Data       ]
      3  REQMENU       MR [Request PC Menu   ]
      4  SENDDATA      MR [Send PC Data      ]
      5  SENDEVNT      MR [Send PC Event     ]
      6  SENDSYSV      MR [Send PC Sysvar    ]
      7  TEST1         PC [TEST PROGRAM1     ]
      8  TEST2         PC [TEST PROGRAM2     ]
      9  KRLPRG        PC [KAREL PROGRAM     ]
     10  ZPRG001       PC [Program001        ]

 [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR]
```

3 Hold dead man switch and turn on TP enable switch. Press [FWD] key with [SHIFT] key pressed.

NOTE
Some KAREL programs keep running even if [SHIFT] key is released after the program is run depending on its attribute. Please confirm details with the author of KAREL program.

NOTE
KAREL program can be selected and run in SELECT screen just like TP program. However, you can not look contents of KAREL program by EDIT screen. If you try to display EDIT screen with KAREL program selected, warning “TPIF-047 Invalid program is selected.” is posted.

9.21.5.2 CALL KAREL program by TP program

Step

1. Press F1, [INST] in EDIT screen.
2. Select CALL and CALL program.
3. Press F3, KAREL. This is displayed only when system variable $KAREL_ENB is 1.

```
SAMPLE1
                                                 4/6
      GEMDATA
      GET_HOME
      KRLPRG                            NT30

 PROGRAM | MACRO | KAREL | INDIRECT | STRINGS | >
```

4. List of programs is displayed. Select KAREL program to be called.

```
SAMPLE1
                                                 4/6
        1: J P[1] 100% FINE
        2: J P[2] 70% CNT50
        3: L P[3] 1000cm/min CNT30
        4: CALL KRLPRG
        5:
     [End]

  [ INST ] |  |  |  | [EDCMD] | >
```

9.21.5.3 Register as MACRO program

Step

1. Press [MENU] key.
2. Select “6 SETUP”.
3. Press F1 to display TYPE menu.
4. Select “MACRO”. Macro screen is displayed.
5. For macro instruction input, pressthe [ENTER] key to display the character string input screen. input characters with function key. Press the [ENTER] key after inputting.
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.
7. For device assignment, press F4, [CHOICE] to display a directory of programs. Then choose a program from the directory.
8. Enter a desired device number.

```
Macro Command
                                                1/150
         Instruction name   Program     Assign
      1 [Program select][STPRGNAM  ] DI[     1]
      2 [              ][          ] --[     0]
      3 [              ][          ] --[     0]
      4 [              ][          ] --[     0]
      5 [              ][          ] --[     0]
      6 [              ][          ] --[     0]
      7 [              ][          ] --[     0]
      8 [              ][          ] --[     0]
      9 [              ][          ] --[     0]
     10 [              ][          ] --[     0]

   [ TYPE ]   CLEAR         [CHOICE]
```

9 Run registered macro program. In the example in figure above, STPRGNAM is run when DI [1] is turned on.

9.21.5.4 Message display by KAREL program

KAREL program may display message in USER screen.

Step

1 Press [MENU] key.
2 Select “9 USER”.
3 USER screen is displayed.
Following figure is an example of message display.

```
USER

Done successfully!
```

9.21.6 KAREL Variable and KAREL Position Variable

You can confirm value of global variable of KAREL program that is selected by SELECT screen.

A Global variable is a variable that is declared in main program. For more detail, please refer to section “Overview of the element of KAREL program” and “Detail of the element of KAREL program” of the “FANUC Robot series R-30*i*A/ R-30*i*A Mate CONTROLLER KAREL Function OPERATOR’S MANUAL” (B-83144EN).
The following is steps to display KAREL variable and KAREL position variable screen.

Step

1 Select KAREL program in SELECT screen.
2 Press DATA.
3 Press F1, [TYPE] and select “KAREL Vars” or “KAREL posns”.

```
DATA Registers
                                          1/200
    R[  1:                  ]=11
    R[  2:                  ]=0
    R[  3:                  ]=0
    R[  4:                  ]=0
  +---------------+         ]=0
  |      TYPE   1 |         ]=0
  |1 Registers    |         ]=0
  |2 Position Reg |         ]=0
  |3 String Reg   |         ]=0
  |4 KAREL Vars   |         ]=0
  |5 KAREL Posn   |
  +---------------+
   [ TYPE ]
```

Following figure is an example of KAREL variable screen.

```
DATA KAREL Vars
                                       1/5
      1 STATUS             0
      2 DETATYPE           1
      3 RARG               3
      4 IARG               0
      5 STRARAG            ******

   [ TYPE ]
```

Following figure is an example of KAREL position variable screen.

```
DATA KAREL Posn
                                       1/2
      1 JPOS1              G1@Recorded
      2 JPOS2              G2 Not Recorded

   [ TYPE ]  DETAIL  RECORD  MOVE_LN  MOVE_JT
```

9.22 SOFT OPERATOR PANEL

9.22.1 OVERVIEW

Using Soft Operator Panel you can start production from a PC screen instead of the hardware operator panel on the controller. To use this function, *i*RProgrammer option (J767) or Remote *i*Pendant option (R843) is needed. For R-30*i*B Compact Plus and R-30*i*B Mini Plus, this option is provided by default.

The Soft Operator Panel is used to start the execution of robot programs from a PC. When the Soft Operator Panel is enabled, production cannot be started using any other method.

The Soft Operator Panel can open from only one PC. The IP address of the PC which uses Soft Operator Panel has to be defined in the controller. Please refer to “9.22.2 SETUP” for more information on setup.

The Soft Operator Panel provides the following functions.

- Cycle start by soft key

- Hold and Abort All by soft keys
- Reset alarm by soft key
- Display program status
- Change Remote/Local setting

Please refer to “9.22.3 OPERATION” for more detail.

These functions work on a web browser. Installing any additional software is not required.
The following browsers are supported.

- Google Chrome (after version 31)
- Internet Explorer 11

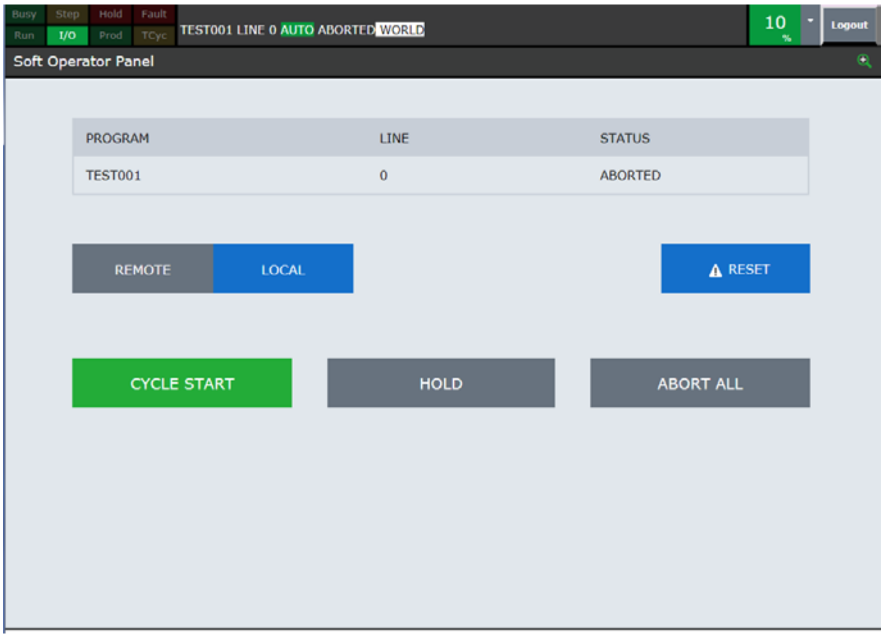


Fig. 9.22.1(a) Soft Operator Panel Screen

> ⚠ **WARNING**
> 1 When Soft Operation Panel is used, a risk assessment for the whole robot system is necessary to verify Soft Operation Panel is adequate.
> 2 To use this function, an emergency stop button must be connected and located near the operator for use at any time.

An emergency stop button has to be connected to the external emergency stop input. Please refer to the following any of manuals to see how to connect to the external emergency stop input.

- FANUC Robot series R-30*i*B/R-30*i*B Plus CONTROLLER MAINTENANCE MANUAL (B-83195EN) Subsection 3.3.5 “Connecting the External Emegency Stop” in Part II “CONNECTIONS”
- FANUC Robot series R-30*i*B Mate/R-30*i*B Mate Plus CONTROLLER MAINTENANCE MANUAL (B-83525EN) Subsection 3.3.4 “Connecting the External Emegency Stop” in Part II “CONNECTIONS”
- FANUC Robot series R-30*i*B Mate/R-30*i*B Mate Plus CONTROLLER Open Air MAINTENANCE MANUAL (B-83555EN) Subsection 1.3 “CONNECTING THE EXTERNAL EMERGENCY STOP” in Part II “CONNECTIONS”
- FANUC Robot series R-30*i*B Compact Plus CONTROLLER MAINTENANCE MANUAL (B-84035EN) Subsection 2.5 “CONNECTING THE EXTERNAL EMERGENCY STOP” in Part II “CONNECTIONS”

- FANUC Robot series R-30*i*B Mini Plus CONTROLLER MAINTENANCE MANUAL (B-84175EN) Subsection 2.5 “CONNECTING THE EXTERNAL EMERGENCY STOP” in Part II “CONNECTIONS”

9.22.2 SETUP

To connect Soft Operator Panel to your PC, you must set the TCP/IP setting on the robot controller.

```
SETUP Host Comm
 TCP/IP                                    1/42
     Robot name:                          SCARA
     Port#1 IP addr:              192.168.1.100
     Subnet Mask:                 255.255.255.0
     Board address:           *****************
     Router IP addr:                192.168.1.1
     PC JOG IP addr:                          *
     OP Panel IP addr:             192.18.1.200

   [ TYPE ]   DHCP     PORT     PING     HELP
```

In order to connect to the PC, the IP address of the controller and the PC should be set up.
The IP address of the controller is set in the Host Comm screen.
Soft Operator panel can be used from only one PC. The IP address of the PC has to be set as “OP Panel IP addr” in Host Comm screen. PCs which have other IP addresses cannot connect to the Soft Operator Panel.

To enable the Soft Operator Panel, “Use Soft Operator Panel” setting has to be changed to TRUE on the system configuration screen.

```
System/Config
                                             1/66
   58 Export diagnosis data:          <*DETAIL*>
   59 Check Sim. I/O when FWD/BWD:        FALSE
   60 Confirmation for AUTO:            DI[    0]
   61 Ambient Temperature:            <*DETAIL*>
   62 Allow I/O chg. in servo-off:        TRUE
   63 Prohibit I/O change anytime:        FALSE
   64 Incremental Jog:                    FALSE
   65 Zero 0% Override:                   FALSE
   66 Use Soft Operator Panel:            FALSE

   [ TYPE ]                   TRUE     FALSE
```

If the setting is FALSE, Soft Operator Panel cannot be displayed on any PC.
If the setting is TRUE, Soft Operator Panel is enabled, but the cycle start button on controller is disabled. A program, therefore, does not start by pushing the button.

9.22.3 OPERATION

Soft Operator Panel can be displayed from the robot Homepage. When the "Soft Operator Panel" icon is selected, the Soft Operator Panel is displayed in another window.

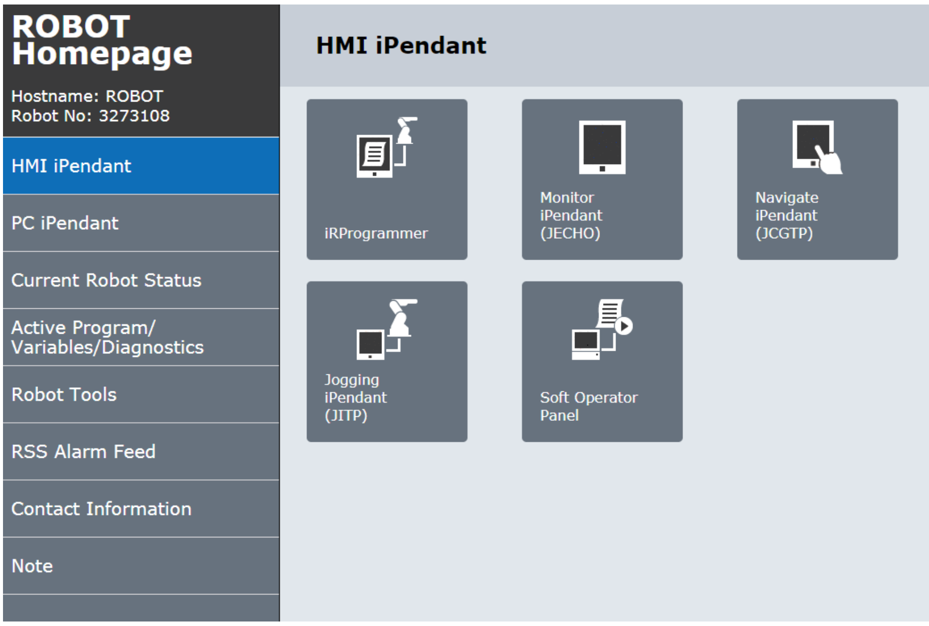


Fig. 9.22.3(a) Robot Home Page

Procedure 9-32 Connection to Soft Operation Panel

Conditions

1. One of the following conditions has to been satisfied:
   - Ethernet port 1 of the controller is connected directly to the PC by an Ethernet cable.
   - The PC and the robot controller are connected to a network.
2. The IP addresses of the PC and controller are both properly set for connectivity, i.e. their IP addresses are set to conform to the network.
3. The IP address of the PC is set as "OP Panel IP addr".
4. "Use Soft Operator Panel" setting is TRUE.

Please refer to "9.22.2 SETUP" for setup information.

Steps

1. The *i*Pendant or Remote *i*Pendant is used to finish the startup after a cycle power. Select a program if none are selected.
2. Activate your browser on your PC.
3. In the Address field of the browser, input "http://<Robot IP Address>" to open the robot Homepage. If the connection is successfully made, you will briefly see the robot homepage, similar to that shown in Fig. 9.22.3(a), displayed on the PC.
4. Select the "Soft Operator Panel" link from the HMI *i*Pendant section.

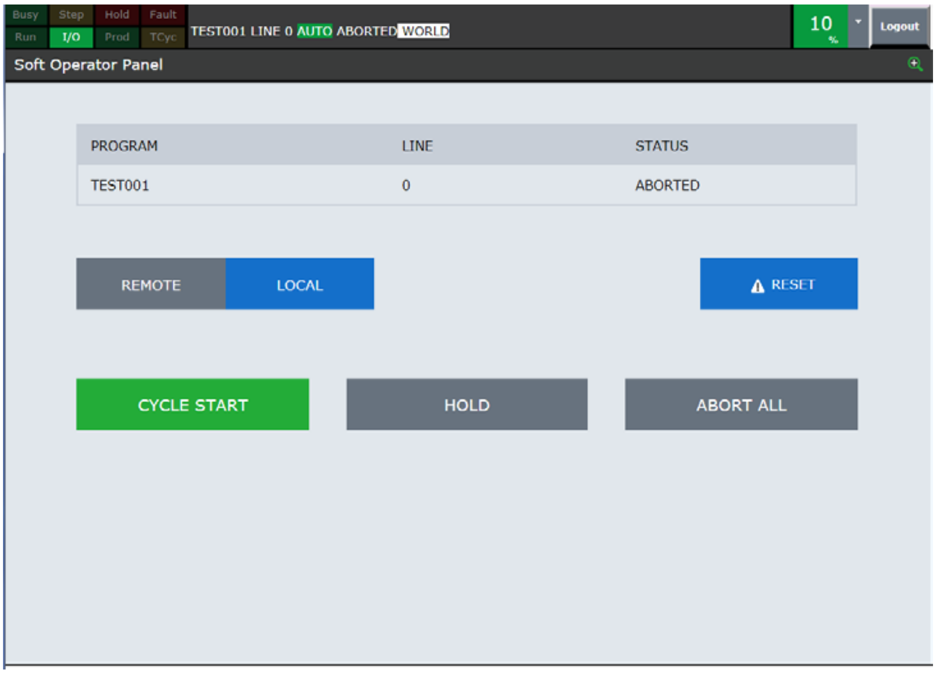


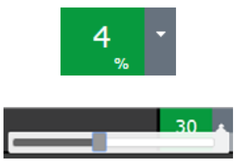
Fig. 9.22.3 (b) Soft Operator Panel Screen

Soft Operator Panel is opened by selecting the “Soft Operator Panel” link in the robot homepage. When the connection is successfully made, you will see the screen, similar to that shown in Fig. 13.4.2(b), displayed on the browser on your PC.

In any of the following situations, the Soft Operator Panel cannot be opened.
- Mode is not AUTO.
- The IP address of the PC is not same as the “OP Panel IP addr” setting in Host Comm screen.
- “Use Soft Operator Panel” setting is FALSE.
- *i*Pendant nor Remote *i*Pendant has not been connected since the last cycle power.

Function of each item on the Soft Operator Panel is described in Table 9.22.2(a).

Table 9.22.2 (a) items on the Soft Operator Panel

Items	Description
PROGRAM	Selected program is displayed.
LINE	Current selected line is displayed.
STATUS	Current status is displayed. ABORTED: The program is aborted. PAUSE: The program is paused. When CYCLE START button is pushed, the program is resumed from the current line. RUNNING: The program is running.
	Slider to change the override value. The slider is displayed when an arrow button is clicked.
	Reset alarm.

Items	Description
REMOTE / LOCAL	Remote/Local setting. Remote: Start program by UOP signals. Local: Start program by pushing CYCLE START button in the Soft Operator Panel.
CYCLE START	Start running or resume the program specified on the status bar.
HOLD	Pause the running program.
ABORT ALL	Abort all running programs.

When connection between the PC and controller is lost, functions in the Soft Operator Panel do not work. In this case, the pop-up that shown in Fig. 9.22.3(c) will be appeared. The pop-up indicates the error status. If you want to reconnect, please tap “OK”. If you tap “CANCEL”, the window will be closed.

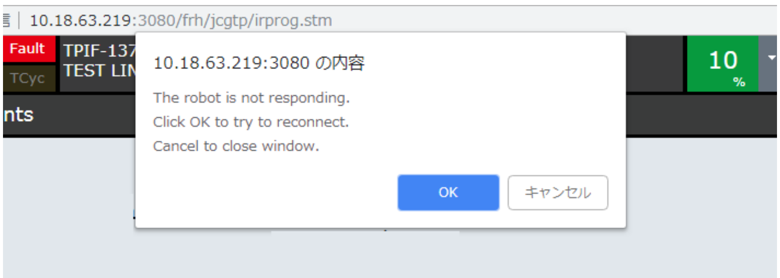


Fig. 9.22.3 (c) Pop-up when connection is lost

Procedure 9-33 Cycle start from Soft Operation Panel

Conditions

1 Mode is AUTO and the TP is disabled.
2 Verify no one is inside the safety fence.
3 Verify the selected program and line number is correct.

Steps

1 Change the Remote/Local setting to Local.
2 Push the Cycle Start key.
3 Confirmation dialog box will be displayed. Select “Yes” to start the program.

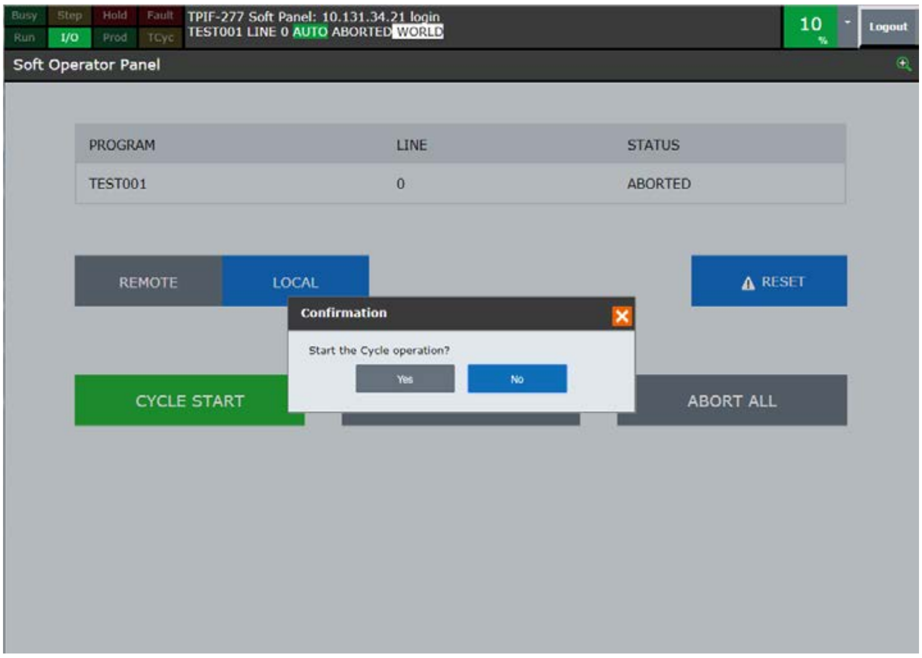


Fig. 9.22.3 (d) Confirmation dialog box to start the program

⚠ **WARNING**

When you start running programs using the Soft Operator Panel, you must strictly adhere to the following precautions.

- Running programs must be executed from outside of the safeguarded space.
- You must check that no one is inside the safety fence.
- Verify that the selected program and line number is correct before running a program.
- An Emergency stop button must be connected and located near the operator for use at any time.

10 PALLETIZING FUNCTION

This chapter explains the palletizing function. To use this function, Palletizing option (J500) is required.

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.

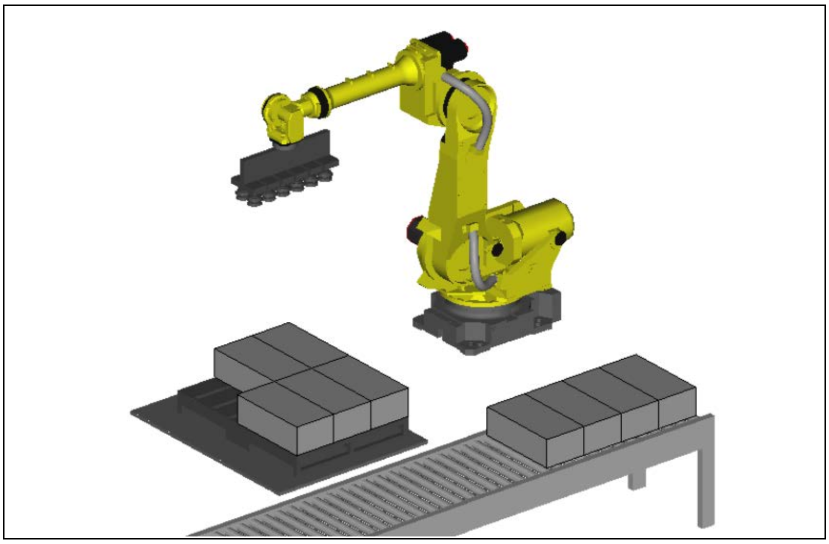


Fig. 10.1 (a) Palletizing

Structure of the palletizing function

The palletizing function consists of the following two patterns:

- Stacking pattern : Determines the method of stacking workpieces.
- Path pattern : Determines the path along which the robot hand moves to stack workpieces.

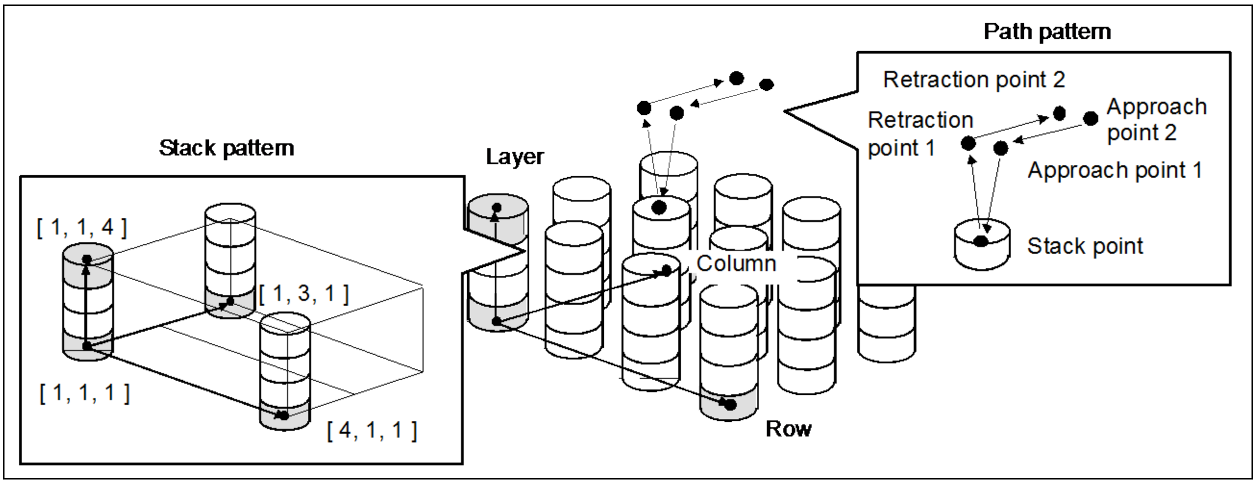


Fig. 10.1 (b) Palletizing pattern

Types of palletizing

There are the following four types of palletizing according to the methods for setting stack and path patterns (See Section 10.3).

- Palletizing B and palletizing BX
- Palletizing E and palletizing EX

- Palletizing B

Palletizing B can achieve a comparatively easy stacking pattern by one kind of path pattern.

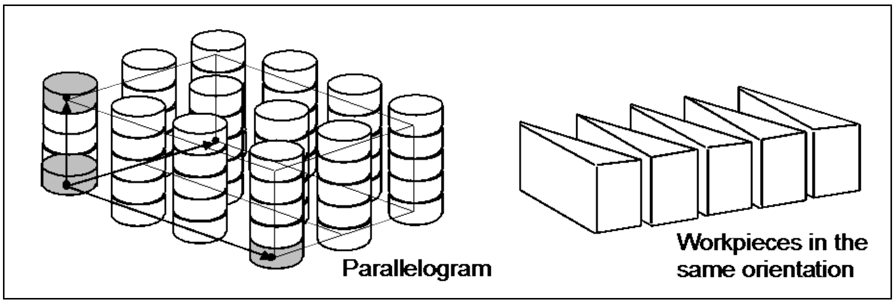


Fig. 10.1 (c) Palletizing-B function

- Palletizing E

Palletizing E can be used for more complex stack patterns (such as when the postures of workpieces are to be changed or when the shape made by the stacked workpieces, as viewed from below, is not a rectangle).

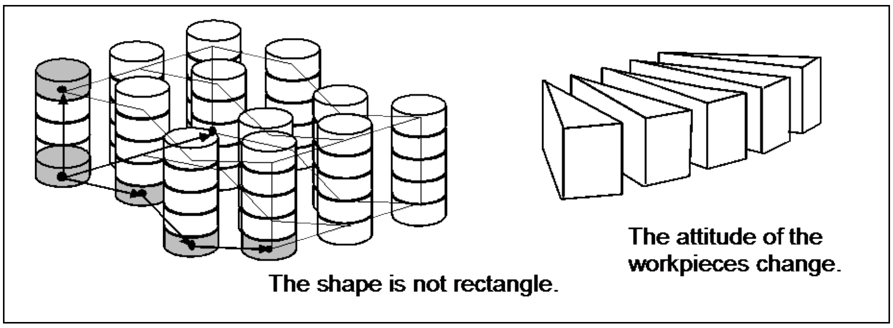


Fig. 10.1 (d) Palletizing E

- Palletizing BX and EX

For palletizing BX and EX, multiple path patterns can be set. For palletizing B and E, only one path pattern can be set.

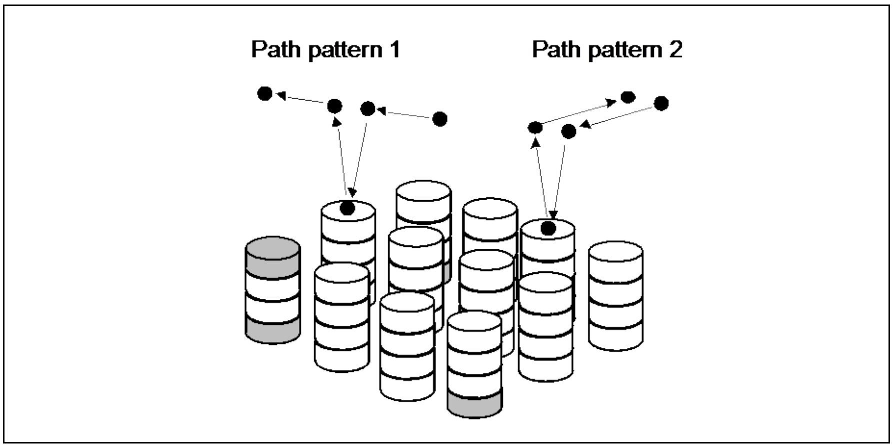


Fig. 10.1 (e) Palletizing BX and EX

10.2 PALLETIZING INSTRUCTIONS

The following palletizing instructions are available:

Table 10.2 Palletizing instructions

Instruction	Function
Palletizing instruction	Calculates the current path based on a stacking pattern, path pattern, and the value held in the palletizing register, and rewrites the position data of a palletizing motion instruction.
Palletizing motion instruction	A motion instruction dedicated to palletizing. It has position data of an approach point, stack point, or retraction point.
Palletizing end instruction	Increments (or decrements) the value of a palletizing register.

Palletizing instruction

Based on the value held in the palletizing register, the palletizing instruction calculates the position of the current stack point from a stack pattern, and also calculates the current path from a path pattern. It then rewrites the position data of a palletizing motion instruction.

```
PALLETIZING-[pattern]_i
     B, BX, E, EX ┘    └ Palletizing number (1 to 16)
```

Fig. 10.2 (a) Format of the palletizing instruction

Palletizing motion instruction

The palletizing motion instruction is a motion instruction that uses three path points ─ an approach point, stack point, and retraction point ─ as position data. This instruction is dedicated to palletizing. Each palletizing instruction rewrites such position data.

```
J   PAL_i   [A_1]   100%   FINE
Palletizing number ┘  └Path point
   (1 to 16)            A_n   : Approach point, n = 1 to 8
                        BTM   : Stack point
                        R_n   : Retraction point, n = 1 to 8
```

Fig. 10.2 (b) Format of the palletizing motion instruction

Palletizing end instruction

The palletizing end instruction calculates the next stack point and increments (or decrements) the palletizing register value.

```
PALLETIZING-END_i
                └─ Palletizing number (1 to 16)
```

Fig. 10.2 (c) Format of the palletizing end instruction

Example

```
1:   PALLETIZING-B_3
2: J PAL_3[ A_2 ] 50% CNT50
3: L PAL_3[ A_1 ] 100mm/sec CNT10
4: L PAL_3[ BTM ] 50mm/sec FINE
5:   hand1 open
6: L PAL_3[ R_1 ] 100mm/sec CNT10
7: J PAL_3[ R_2 ] 100mm/sec CNT50
8:   PALLETIZING-END_3
```

- Palletizing number

Upon completion of the teaching of palletizing data, palletizing numbers are written automatically together with the instructions (palletizing instruction, palletizing motion instructions, and palletizing end instruction).

When a new palletizing operation is taught, a palletizing number is assigned automatically.

Palletizing register instruction

The palletizing register instruction is used to control palletizing. It performs stack point specification, comparison, and branch. (See Subsection 10.4.1.)

```
                     PL[ i ] = (value)
Palletizing register ─┘       ├─ PL[ i ]     : Pallet register[ i ]
number (1 to 32)              └─ [ i , j , k ] : Palletizing register element
```

Fig. 10.2 (d) Palletizing register

10.3 TEACHING THE PALLETIZING FUNCTION

The palletizing function is taught using the following procedure:

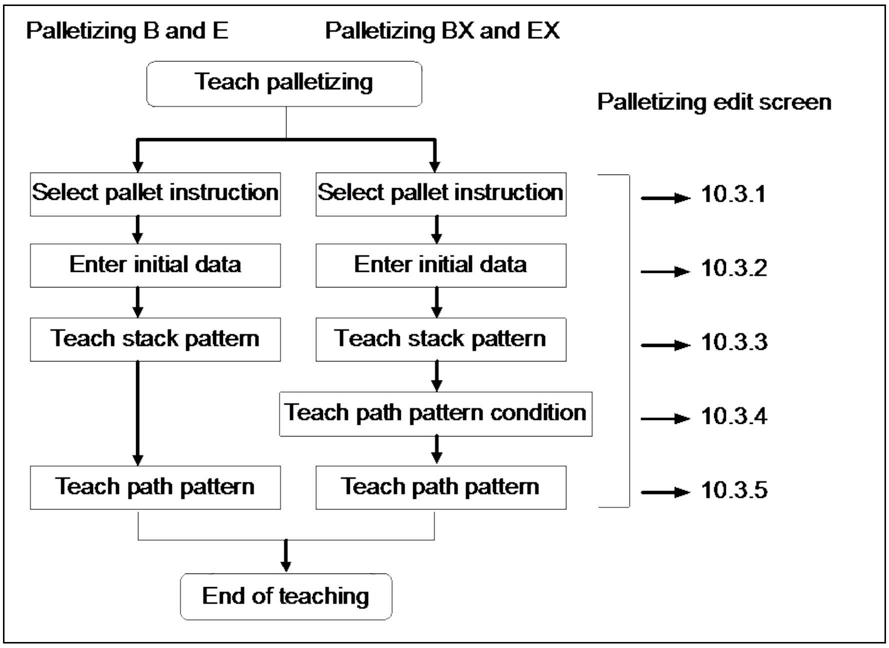


Fig. 10.3 Procedure for teaching the palletizing function

The palletizing function is taught on the palletizing edit screens. One of the palletizing edit screens will be displayed automatically when a palletizing instruction is selected.
When the palletizing function is taught, necessary palletizing instructions such as a palletizing instruction, palletizing motion instruction, and palletizing end instruction are inserted automatically.
The following sections explain the teaching of palletizing EX. For palletizing B, BX, or E, assume that some functions of palletizing EX are restricted.

> **NOTE**
> To improve the motion accuracy of palletizing, Tool Center Point should be accurately set. (See Subsection 3.9.1,"Setting a Tool Coordinate System".)

10.3.1 Selecting a Palletizing Instruction

To select a pallet instruction, select the type of palletizing to be taught (palletizing B, BX, E, or EX).

Procedure 10-1 Selecting a palletizing instruction

Condition

- Make sure that the teach pendant is enabled.
- Make sure that the palletizing instruction is selected on the program edit screen.

```
PROGRAM1
                                            6/6
     5: L  P[2] 300mm/sec CNT50
   [End]

 | POINT |       |       |       | TOUCHUP | >
```

Step

1 Press NEXT “>” to display the next page. Press F1, [INST] to display a menu.

```
PROGRAM1
                                          6/6
       5: L         Instruction   3
  Instruction    2 NDFOR
 Instruction    1 ellaneous Offset
 1 REGISTER       ents      PREG
 2 I/O                      OR/MON.
 3 IF/SELECT      oad
 4 WAIT           et/Frames GS
 5 JMP/LBL        iple
 6 CALL           l
 7 Palletizing    ram
 8 --next page--

   POINT                       TOUCHUP   >
```

2 Select “Palletizing”.

```
PROGRAM1
                                          6/6
       5: L  P[2] 300mm/sec CNT 50
 PALLETIZING statement   1
 1 PALLETIZING-B
 2 PALLETIZING-BX
 3 PALLETIZING-E
 4 PALLETIZING-EX
 5 PALLETIZING-END
 6
 7
 8

   POINT                       TOUCHUP   >
```

3 Select “PALLETIZING-EX.” The screen then changes to the initial data input screen automatically, one of the palletizing edit screen.

```
PROGRAM1
     PALLETIZING Configuration

       PALETIZING_1  [                  ]
       TYPE = [PALLET    ]   INCR = [   1 ]
       PAL REG   = [  1] ORDER = [RCL]
         ROWS    = [  1   LINE   FIX ]
         COLUMNS = [  1   LINE   FIX ]
         LAYERS  = [  1   LINE   FIX    1 ]
         AUXILIARY POS = [ NO         ]
       APPR =[ 1] RTRT =[ 1] PATTERN=[ 1]


     Press ENTER

   PROG                        DONE
```

10.3.2 Inputting Initial Data

On the initial data input screen, specify how palletizing is performed. The data set on the initial data input screen is used for subsequent teach screens. The initial data screen contains the following items:

For palletizing B

```
PROGRAM1
    PALLETIZING Configuration

      PALETIZING_4  [WORK PALLET       ]
      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

```
PROGRAM1
    PALLETIZING Configuration

      PALETIZING_4  [WORK PALLET       ]
      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

```
PROGRAM1
    PALLETIZING Configuration

      PALETIZING_4  [WORK PALLET       ]
      TYPE = [PALLET    ]   INCR = [   1 ]
      PAL REG   = [  1] ORDER = [RCL]
        ROWS    = [  5   LINE   FIX ]
        COLUMNS = [  4   LINE   FIX ]
        LAYERS  = [  3   LINE   FIX    1 ]
        AUXILIARY POS = [ NO         ]
      APPR = [ 2] RTRT = [ 2]


    Press ENTER

   PROG                          DONE
```

For palletizing EX

```
PROGRAM1
    PALLETIZING Configuration

      PALETIZING_4  [WORK PALLET       ]
      TYPE = [PALLET    ]   INCR = [   1 ]
      PAL REG   = [  1] ORDER = [RCL]
        ROWS    = [  5   LINE   FIX ]
        COLUMNS = [  4   LINE   FIX ]
        LAYERS  = [  3   LINE   FIX    1 ]
        AUXILIARY POS = [ NO         ]
      APPR = [ 2] RTRT = [ 2] PATTERN= [ 2]


    Press ENTER

   PROG                          DONE
```

Table 10.3.2 (a) Types of palletizing

	Arrangement mode	Layer pattern	Posture control	Path pattern count
B	2-point teaching only	Not set	Always fixed	1
BX	2-point teaching only	Not set	Always fixed	1 to 16
E	2-point teaching, all-point teaching, or interval specification	Set	Fixed or split	1
EX	2-point teaching, all-point teaching, or interval specification	Set	Fixed or split	1 to 16

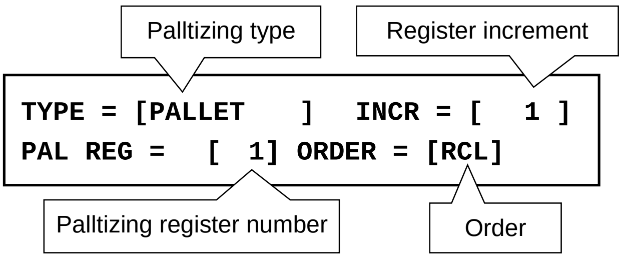
When a pallet instruction is selected, the initial data input screen corresponding to the selected type of palletizing will be displayed. For palletizing EX, all palletizing functions can be specified. For palletizing B, BX, and E, restrictions are imposed on the specification of the functions.
This section explains how to enter initial data for palletizing EX. For palletizing B, BX, or E, assume that some functions of palletizing EX are restricted.

Table 10.3.2 (b) Initial palletizing data

Palletizing number	A number is assigned automatically when a palletizing statement is taught. PALLETIZING_N: 1 to 16
Palletizing type	Specify whether the palletizing register is to be incremented or decremented by the palletizing end instruction. (See Subsection 10.4.1.) Select stacking (PALLET) or unstacking (DEPALLET).
Register increment	Specify the value by which the value held in the palletizing register is to be incremented or decremented by the palletizing end instruction. (See Subsection 10.4.1.)
Palletizing register	Specify the palletizing register to be used by the palletizing instruction and palletizing end instruction.
Order	Specify the stacking (unstacking) order of row, column, and layer. R: Row, C: Column, L: Layer
Numbers of rows, columns, and layers Arrangement mode Posture control Layer pattern count	Numbers of rows, columns, and layers for a stacking pattern. (See Subsection 10.3.3.) 1 to 127 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). Control the posture at rows, columns, and layers for a stacking pattern. Select E or EX. 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 retraction points Path pattern count	Number of approach points in a path pattern. (See Subsection 10.3.5.) 0 to 8 Number of retraction points in a path pattern. (See Subsection 10.3.5.) 0 to 8 Number of path patterns (Subsection 10.3.4) (only for palletizing BX or EX). 1 to 16

Initial data related to the stacking method

In the palletizing function, the stack point is controlled using a palletizing register. (See Subsection 10.4.1, “Palletizing Register”.) How the palletizing register is controlled can be specified as initial data. According to this data, the way of stacking is determined.



- For the pallet type (TYPE), specify either PALLET or DEPALLET (standard setting: PALLET). (See Subsection 10.4.1, “Palletizing Register”.)
- For the register increment (INCR), specify by which amount the stack (unstack) position advances or retracts. That is, specify a value by which the palletizing register is incremented or decremented by the palletizing end instruction. The standard setting is 1. (See Subsection 10.4.1, “Palletizing Register”.)
- As the palletizing register, specify the register number of a palletizing register used for stack control.

> ⚠ **CAUTION**
> Make sure that the specified palletizing register number is not used by another palletizing function.

- For the order (ORDER), specify the stacking/unstacking order of row, column, and layer.

Stacking workpieces in the order:
Row, Column, Layer

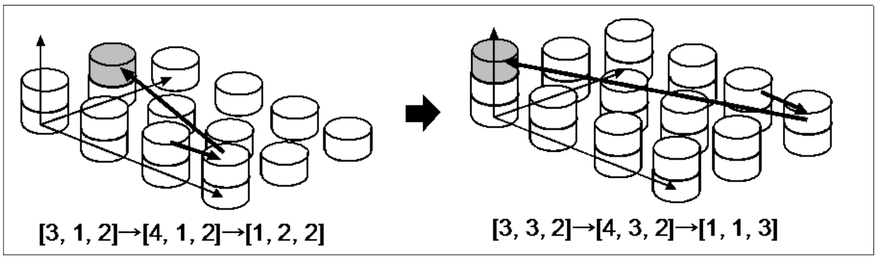


Fig. 10.3.2 Palletizing order

Initial data related to a stacking pattern

As the stacking pattern data, specify the numbers of rows, columns, and layers, arrangement mode, posture control type, the number of layer patterns and also specify whether to provide an auxiliary position. (See Subsection 10.3.3, “Teaching a Stacking Pattern”.)

Number of rows, columns,and layers | Arrangement mode | Posture control

```
ROWS     = [  4   LINE   FIX ]
COLUMNS  = [  3   LINE   FIX ]
LAYERS   = [  5   LINE   FIX   1  ]
AUXILIARY POS = [ YES   FIX  ]
```

Auxiliary position | Posture control at auxiliary position

Initial data related to a path pattern

As the initial path pattern data, specify the number of approach points, the number of retraction points and the number of path patterns. (See Subsection 10.3.5, “Teaching a Path Pattern”.)

```
APPR = [ 2] RTRT = [ 2] PATTERN = [ 1]
```

Number of approach points | Number of retraction points | Number of path patterns

Procedure 10-2 Inputting initial palletizing data

Step

1 As the palletizing instruction, select palletizing-EX. The initial data input screen will be displayed. (See Subsection 10.3.1, “Selecting a Palletizing Instruction”.)

```
PROGRAM1
     PALLETIZING Configuration

       PALETIZING_4  [                  ]
       TYPE = [PALLET    ]   INCR = [   1 ]
       PAL REG   = [  1] ORDER = [RCL]
         ROWS    = [  1   LINE   FIX ]
         COLUMNS = [  1   LINE   FIX ]
         LAYERS  = [  1   LINE   FIX    1 ]
         AUXILIARY POS = [ NO         ]
       APPR =[ 1] RTRT =[ 1] PATTERN=[ 1]


     Press ENTER
   PROG                        DONE
```

NOTE

PALLETIZING_4 displayed on the initial data input screen indicates the 4th 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 menu will be displayed.

```
PROGRAM1
     PALLETIZING Configuration

       PALETIZING_4  [                  ]
       TYPE = [PALLET    ]   INCR = [   1 ]
       PAL REG   = [  1] ORDER = [RCL]
         ROWS    = [  1   LINE   FIX ]
         COLUMNS = [  1   LINE   FIX ]
         LAYERS  = [  1   LINE   FIX    1 ]
         AUXILIARY POS = [ NO    Alpha input  1
       APPR =[ 1] RTRT =[ 1] PATTE  Upper Case
                                    Lower Case
                                    Punctuation
                                    Options
   ABCDEF  GHIJKL  MNOPQR  STUVWX  YZ_@*  >
```

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
     PALLETIZING Configuration

       PALETIZING_4  [WORK PALLET       ]
       TYPE = [PALLET    ]   INCR = [   1 ]
       PAL REG   = [  1] ORDER = [RCL]
         ROWS    = [  1   LINE   FIX ]
         COLUMNS = [  1   LINE   FIX ]
         LAYERS  = [  1   LINE   FIX    1 ]
         AUXILIARY POS = [ NO    Alpha input  1
       APPR =[ 1] RTRT =[ 1] PATTE  Upper Case
                                    Lower Case
                                    Punctuation
                                    Options
   abcde  ghijkl  mnopqr  stuvwx  yz_@*  >
```

3 To select a palletizing type, move the cursor to the TYPE field, then select a function key.

```
PROGRAM1
     PALLETIZING Configuration

       PALETIZING_4  [WORK PALLET       ]
       TYPE = [PALLET    ]   INCR = [   1 ]
       PAL REG    = [  1] ORDER = [RCL]
         ROWS     = [  1   LINE   FIX ]
         COLUMNS = [  1   LINE   FIX ]
         LAYERS  = [  1   LINE   FIX    1 ]
         AUXILIARY POS = [ NO         ]
       APPR =[ 1] RTRT =[ 1] PATTERN=[ 1]

 | PROG | PALLET | DEPALL |        | DONE |
```

4 To enter a numeric value to the item INCR and PAL_REG, press a numeric key, then press the [ENTER] key.

```
PROGRAM1
     PALLETIZING Configuration

       PALETIZING_4  [WORK PALLET       ]
       TYPE = [PALLET    ]   INCR = [   1 ]
       PAL REG    = [  1] ORDER = [RCL]
         ROWS     = [  1   LINE   FIX ]
         COLUMNS = [  1   LINE   FIX ]
         LAYERS  = [  1   LINE   FIX    1 ]
         AUXILIARY POS = [ NO         ]
       APPR =[ 1] RTRT =[ 1] PATTERN=[ 1]

 | PROG |        |        |        | DONE |
```

5 Specify a palletizing order by selecting the function keys in the target order.

```
PROGRAM1
     PALLETIZING Configuration

       PALETIZING_4  [WORK PALLET       ]
       TYPE = [PALLET    ]   INCR = [   1 ]
       PAL REG    = [  1] ORDER = [R__]
         ROWS     = [  1   LINE   FIX ]
         COLUMNS = [  1   LINE   FIX ]
         LAYERS  = [  1   LINE   FIX    1 ]
         AUXILIARY POS = [ NO         ]
       APPR =[ 1] RTRT = [ 1] PATTERN= [ 1]


     Select key
 |        |   R    |   C    |   L    |        |
```

After the second item has been selected, the third item is determined automatically.

```
PROGRAM1
     PALLETIZING Configuration

       PALETIZING_4  [WORK PALLET       ]
       TYPE = [PALLET    ]   INCR = [   1 ]
       PAL REG   = [  1] ORDER = [RCL]
         ROWS    = [  1   LINE   FIX ]
         COLUMNS = [  1   LINE   FIX ]
         LAYERS  = [  1   LINE   FIX    1 ]
         AUXILIARY POS = [ NO         ]
       APPR =[ 1] RTRT = [ 1] PATTERN= [ 1]


     Select key
   PROG      R       C       L      DONE
```

6 To specify the number of rows, columns and layers, press a numeric key, then press the [ENTER] key. To specify the type of posture control, move the cursor to the setting field and select the function key.

```
PROGRAM1
     PALLETIZING Configuration

       PALETIZING_4  [WORK PALLET       ]
       TYPE = [PALLET    ]   INCR = [   1 ]
       PAL REG   = [  1] ORDER = [RCL]
         ROWS    = [  1   LINE   FIX ]
         COLUMNS = [  1   LINE   FIX ]
         LAYERS  = [  1   LINE   FIX    1 ]
         AUXILIARY POS = [ NO         ]
       APPR =[ 1] RTRT = [ 1] PATTERN= [ 1]


     Enter value
   PROG     LINE     FIX             DONE
```

7 To specify the arrangement mode by the interval specification, position the cursor to arrangement mode field and input interval value. (The unit of interval is mm.)

```
PROGRAM1
     PALLETIZING Configuration

       PALETIZING_4  [WORK PALLET       ]
       TYPE = [PALLET    ]   INCR = [   1 ]
       PAL REG   = [  1] ORDER = [RCL]
         ROWS    = [  1   200   FIX ]
         COLUMNS = [  1   LINE   FIX ]
         LAYERS  = [  1   LINE   FIX    1 ]
         AUXILIARY POS = [ NO         ]
       APPR =[ 1] RTRT = [ 1] PATTERN= [ 1]


     Enter value
   PROG     LINE     FIX             DONE
```

8 To specify whether to set auxiliary points, position the cursor at the auxiliary point field and select the desired function key menu.

```
PROGRAM1
     PALLETIZING Configuration

       PALETIZING_4  [WORK PALLET       ]
       TYPE = [PALLET    ]   INCR = [   1 ]
       PAL REG   = [  1] ORDER = [RCL]
         ROWS    = [  1   200   FIX ]
         COLUMNS = [  1   LINE   FIX ]
         LAYERS  = [  1   LINE   FIX    1 ]
         AUXILIARY POS = [ NO         ]
       APPR =[ 1] RTRT = [ 1] PATTERN= [ 1]


     Select key
    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
     PALLETIZING Configuration

       PALETIZING_1  [WORK PALLET       ]
       TYPE = [PALLET    ]   INCR = [   1 ]
       PAL REG   = [  1] ORDER = [RCL]
         ROWS    = [  4   LINE   FIX ]
         COLUMNS = [  3   LINE   FIX ]
         LAYERS  = [  5   LINE   FIX    1 ]
         AUXILIARY POS = [ NO         ]
       APPR =[ 2] RTRT =[ 2] PATTERN=[ 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 will be displayed.

```
PROGRAM1
     PALLETIZING Bottom Points

        1: *P[  1,  1,  1]               1/1
        2: *P[  4,  1,  1]
        3: *P[  1,  3,  1]
        4: *P[  1,  1,  5]
     [End]

    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 Subsection 10.4.1, “Palletizing Register”.)

10.3.3 Teaching a Stacking Pattern

On the palletizing stacking pattern teach screen, teach representative stack points of a stacking pattern. From these representative points, a target stack point is calculated automatically at the time of palletizing.

Stacking pattern teach screen

```
PROGRAM1
     PALLETIZING Bottom Points

        1:--P[  1,  1,  1]--              1/1
        2:--P[  4,  1,  1]--
        3:--P[  1,  3,  1]--
        4:--P[  1,  1,  5]--
     [End]

   |  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.

With or without an auxiliary position

For the stacking pattern without an auxiliary position, individually teach four tops of the stacking pattern.

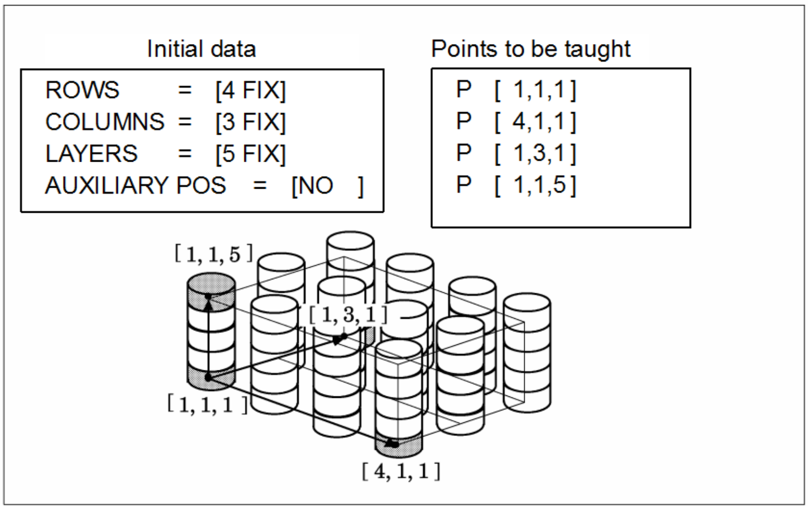


Fig. 10.3.3 (a) Stacking pattern with no auxiliary position

For a stacking pattern with an auxiliary position, when the shape of the first layer is a trapezoid, also teach the fifth position using the function provided.

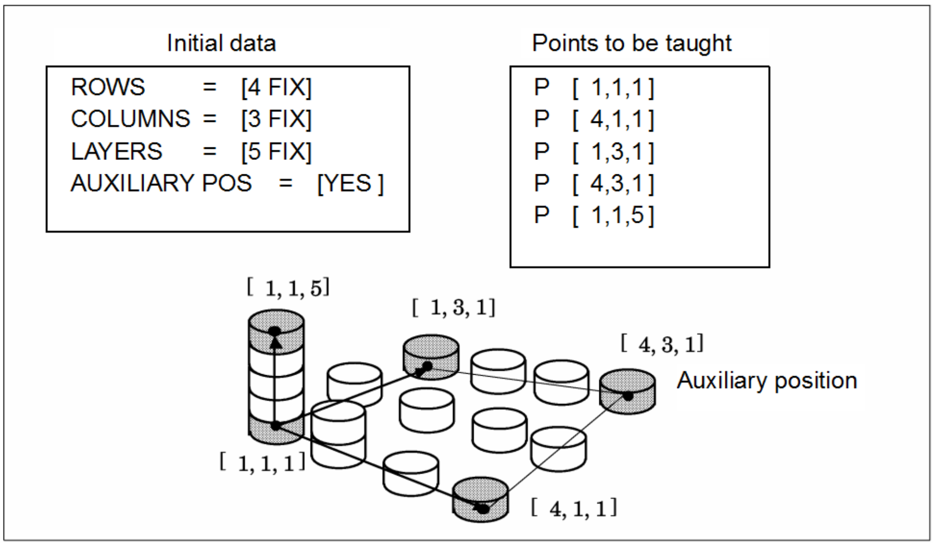


Fig. 10.3.3 (b) Stacking pattern with an auxiliary position

When selecting with auxiliary point, specify posture control type (fixed, split) at the auxiliary point. (E, EX only)

Types of arrangement modes/2-point teaching

When 2-point teaching is selected, teach the representative two points at both ends to set all the points in the row, column, and layer directions (standard).

> **NOTE**
> The following explanation is not relevant to palletizing B and BX. See Subsection 10.3.4.

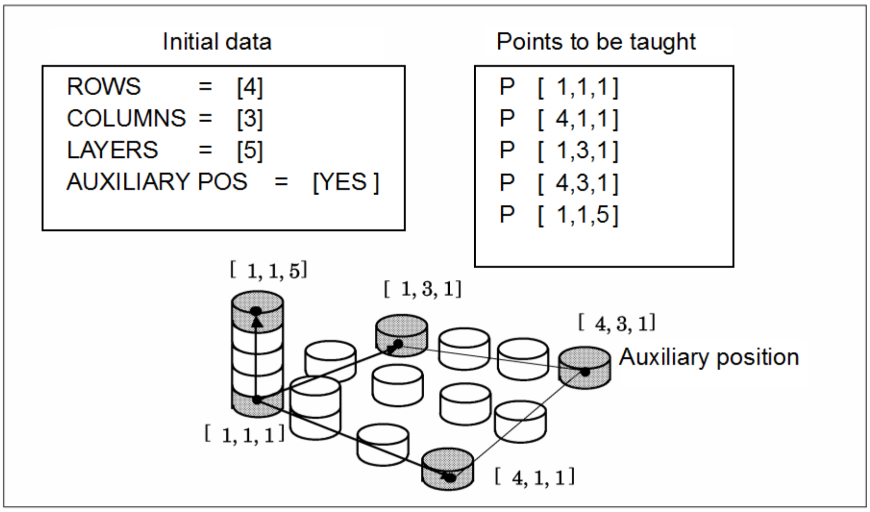


Fig. 10.3.3 (c) Teaching method by 2-point teaching

All-point teaching

When selecting all-point teaching, directly teach all the points in the row, column, and layer directions.

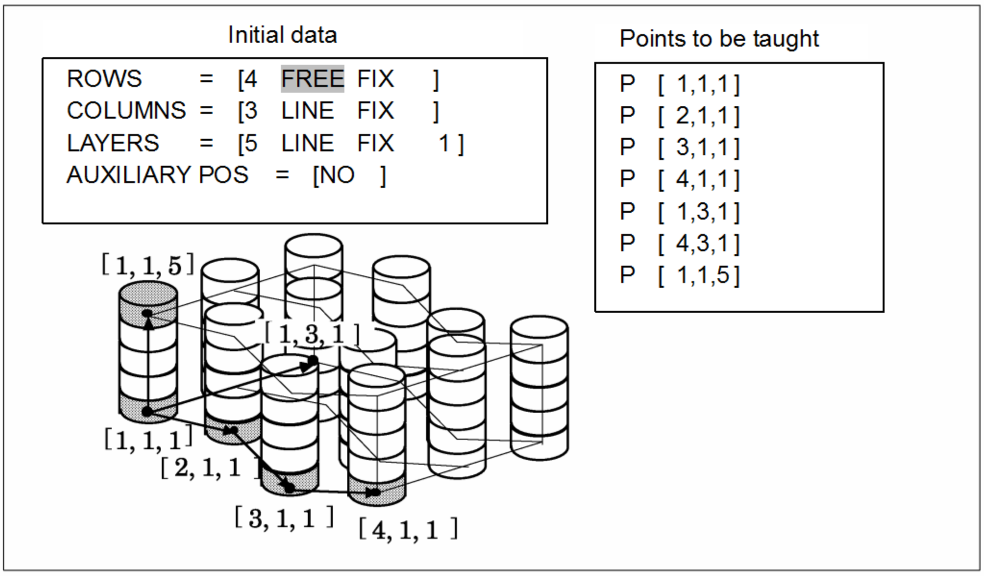


Fig. 10.3.3 (d) Teaching method by all-point teaching

Interval specification

When selecting the interval specification, specify the two points at both ends in each of the row, column, and layer directions, as well as the distance between workpieces, to set all points.

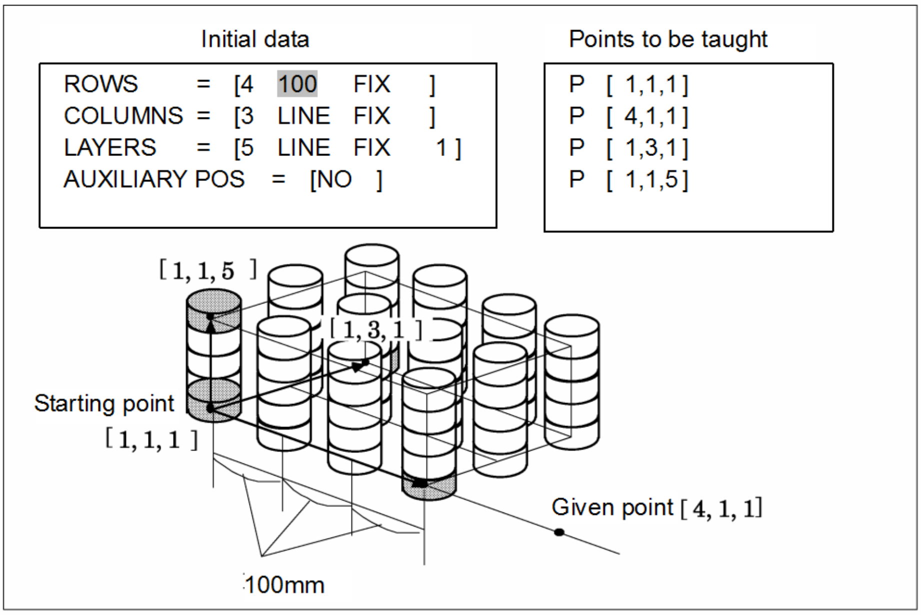


Fig. 10.3.3 (e) Teaching method by interval specification

Types of posture control

When the fixed posture is specified, workpieces at all the stack points always take the posture taught at point [1,1,1] (standard).

Initial data

```
ROWS     = [5  LINE  FIX    ]
COLUMNS  = [1  LINE  FIX    ]
LAYERS   = [1  LINE  FIX   1 ]
AUXILIARY POS  =  [NO  ]
```

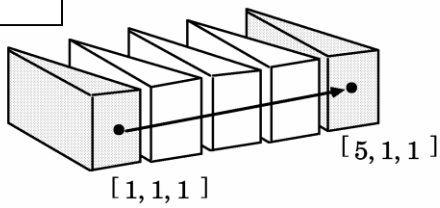


Fig. 10.3.3 (f) Postures of workpieces at stack points when the fixed posture is specified

For the split posture, when 2-point teaching is specified, workpieces take the postures obtained by splitting the postures taught at the two end points. When all-point teaching is specified, workpieces take the postures at the taught points.

Initial data

```
ROWS     = [5  LINE  INTER]
COLUMNS  = [1  LINE  FIX    ]
LAYERS   = [1  LINE  FIX   1 ]
AUXILIARY POS  =  [NO  ]
```

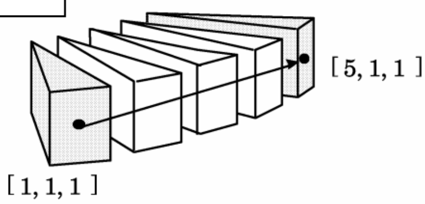


Fig. 10.3.3 (g) Postures of workpieces at stack points when the split posture is specified

Layer pattern count

To change the stack pattern every few layers, enter the number of layer patterns.
The layer pattern count is valid only when 2-point teaching is specified for the layer arrangement (for other cases, the layer pattern count is always 1).
For the first layer, the workpieces are always stacked at the stack points in layer pattern 1.
When the layer pattern count is N, the numbers of layers and layer patterns are the same until layer N. For layer (N+1) and beyond, layer patterns starting from layer pattern 1 are repeated.
Teach the position for the layer direction only in the teaching for layer pattern 1.
The position for the layer direction in each layer is calculated by the taught position for layer pattern1.

```
Initial data
ROWS      =  [2  FREE FIX     ]
COLUMNS   =  [2  FREE FIX     ]
LAYERS    =  [4  LINE  FIX   2]
AUXILIARY POS  =  [NO  ]
```

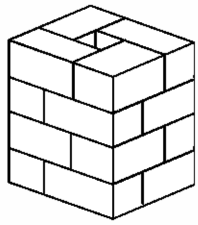


Fig. 10.3.3 (h) Posture at the stacking point in division posture

When the total number of layers is less than 16, a number not greater than the total number can be set for the layer pattern count. If a number less than the layer pattern count is subsequently specified for the number of layers, the layer pattern count is automatically changed to the number of layers.

Procedure 10-3 Teaching a palletizing stacking pattern

Step

1 Based on the initial data setting, a list of stack points to be taught is displayed.

```
PROGRAM1
     PALLETIZING Bottom Points

        1: *P [  1,  1,  1]               1/1
        2: *P [  4,  1,  1]
        3: *P [  1,  3,  1]
        4: *P [  1,  1,  5]
     [End]

  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 [SHIFT] key. The current robot position is then recorded.

```
PROGRAM1
     PALLETIZING Bottom Points

        1:--P [  1,  1,  1]--           1/1
        2: *P [  4,  1,  1]
        3: *P [  1,  3,  1]
        4: *P [  1,  1,  5]
     [End]

  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
     PAL_1[BTM]UF:0   UT:1      CONF:NUT 000
     X   1827.000   mm   W   -180.000   deg
     Y      0.000   mm   P    -90.000   deg
     Z   1300.000   mm   R      0.000   deg
    Position Detail
    PALLETIZING Bottom Points
       1:--P [  1,  1,  1]--            1/1
       2: *P [  4,  1,  1]
       3: *P [  1,  3,  1]
       4: *P [  1,  1,  5]
    [End]

    Enter value

                   CONF   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
     PALLETIZING Bottom Points

        1:--P [  1,  1,  1]--            1/1
        2:--P [  4,  1,  1]--
        3:--P [  1,  3,  1]--
        4:--P [  1,  1,  5]--
     [End]

  BACK              RECORD   DONE   >
```

5 Pressing [FWD] key while holding down [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) (Subsection 10.3.4 or 10.3.5).

> **NOTE**
> When layer patterns are used (E or EX) and F5, DONE is pressed, the screen for specifying the stack pattern for the next layer will be displayed.

```
PROGRAM1
     PALLETIZING Bottom Points

        1:--P [  1,  1,  2]--            2/2
        2:--P [  4,  1,  2]--
        3:--P [  1,  3,  2]--
     [End]

  BACK              RECORD   DONE   >
```

10.3.4 Setting Path Pattern Conditions

The screen explained in this section is not displayed for palletizing B or E. See Subsection 10.3.5.
The palletizing path pattern condition setting screen is used to set conditions indicating which path pattern is to be used for each stack point in advance when multiple path patterns are to be set on the path pattern teaching screen (See Subsection 10.3.5).
For palletizing BX and EX, multiple path patterns can be set independently for stack points. For palletizing B and E, this screen is not displayed because only one path pattern can be set.

Path pattern condition setting screen

```
PROGRAM1
     PALLETIZING Route Patterns

      PTN [ 1] = [     * ,  3- 1 ,     * ]
      PTN [ 2] = [     * ,  3- 2 ,     * ]
      PTN [ 3] = [     * ,  3- 0 ,     * ]
     [End]


     Enter value
```

	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.

In this example, pattern 1 is used for the stack points in column 1, pattern 2 is used for the stack points on column 2, and pattern 3 is used for the stack points in column 3.

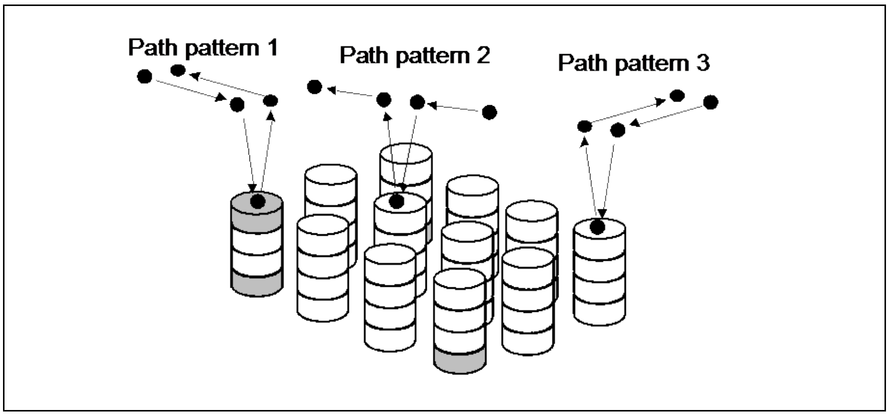


Fig. 10.3.4 (a) Palletizing using three path patterns

How to use path pattern conditions

- The execution of palletizing uses the path pattern with the condition number for which the row, column, and layer numbers at the stack point match the row, column, and layer (element) values of a path pattern condition.
- In direct specification mode, specify numbers from 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] = [* , * , *]

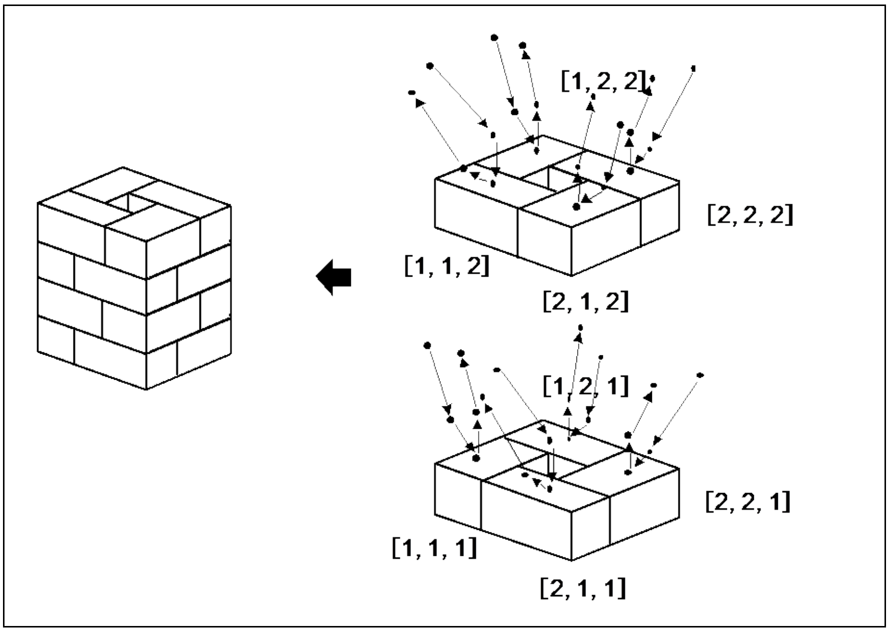


Fig. 10.3.4 (b) Box palletizing using eight path patterns

Example
PTN [1] = [1 , 1 , 2–1]
PTN [2] = [2 , 1 , 2–1]
PTN [3] = [1 , 2 , 2–1]
PTN [4] = [2 , 2 , 2–1]
PTN [5] = [1 , 1 , 2–0]
PTN [6] = [2 , 1 , 2–0]
PTN [7] = [1 , 2 , 2–0]
PTN [8] = [2 , 2 , 2–0]

In this example, eight path patterns are defined and repeated for every two layers because different paths must be set according to the box position.

Procedure 10-4 Setting palletizing path pattern conditions

Step

1 Condition items to be entered are displayed according to the value set for the pattern count as initial data.

```
PROGRAM1
     PALLETIZING Route Patterns

      PTN [ 1] = [     * ,      * ,      * ]
      PTN [ 2] = [     * ,      * ,      * ]
      PTN [ 3] = [     * ,      * ,      * ]
     [End]


     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.

```
PROGRAM1
     PALLETIZING Route Patterns

      PTN [ 1] = [     * ,      1 ,      * ]
      PTN [ 2] = [     * ,      * ,      * ]
      PTN [ 3] = [     * ,      * ,      * ]
     [End]


     Enter value
   | 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.

```
PROGRAM1
     PALLETIZING Route Patterns

      PTN [ 1] = [     * ,    1- 0,      * ]
      PTN [ 2] = [     * ,      * ,      * ]
      PTN [ 3] = [     * ,      * ,      * ]
     [End]


     Enter value
   | BACK |      | DIRECT | MODULO | DONE | >
```

4 To specify values in direct specification mode, press F3, DIRECT.
5 Press F1, BACK to display the previous stack point teaching screen again.
6 Press F5, DONE to display the next path pattern teaching screen.

```
PROGRAM1
    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
    [End]


    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
    PALLETIZING Route Points

          IF PL[ 1]=[*,*,*]            1/3
       1:Joint *P [A_2 ] 30%      FINE
       2:Joint *P [A_1 ] 30%      FINE
       3:Joint *P [BTM ] 30%      FINE
       4:Joint *P [R_1 ] 30%      FINE
       5:Joint *P [R_2 ] 30%      FINE
    [End]


    Teach Route Points
   | BACK | POINT |      | RECORD | DONE | > |
```

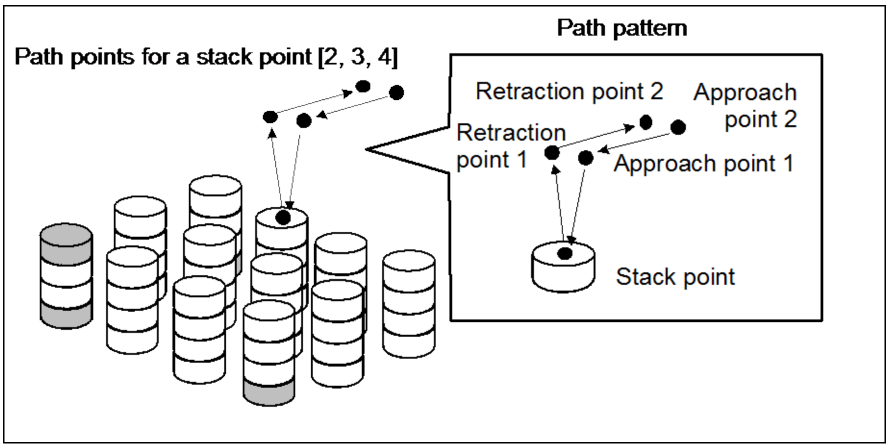


Fig. 10.3.5 Palletizing path

Procedure 10-5 Teaching a palletizing path pattern

Step

1 Based on the initial data setting, a list of the path points to be taught will be displayed.

```
PROGRAM1
    PALLETIZING Route Points

          IF PL[ 1]=[*,*,*]             1/3
       1:Joint *P [A_2 ] 30%       FINE
       2:Joint *P [A_1 ] 30%       FINE
       3:Joint *P [BTM ] 30%       FINE
       4:Joint *P [R_1 ] 30%       FINE
       5:Joint *P [R_2 ] 30%       FINE
    [End]


    Teach Route Points

      BACK    POINT           RECORD   DONE    >
```

> **NOTE**
> The number of path points to be recorded depends on the number of approach points and the number of retraction points set on the initial data input screen. In this sample screen, the number of approach points is 2 and the number of retraction points is 2.

2 Move the robot by jog feed to a position which is to be taught as a path point.

3 Move the cursor to the setting field to be taught and teach the position using one of the following operations.

a Press F2, POINT while holding down [SHIFT] key. When pressing F2, POINT, without pressing and holding [SHIFT] key, default logical motion menu will be displayed and then you can set the motion type or feed rate, etc. (This key is displayed only at teaching route pattern 1.)

b Press and hold [SHIFT] key and press F4, RECORD.

```
PROGRAM1
    PALLETIZING Route Points

          IF PL[ 1]=[*,*,*]             1/3
       1:Joint  P [A_2 ] 30%       FINE
       2:Joint *P [A_1 ] 30%       FINE
       3:Joint *P [BTM ] 30%       FINE
       4:Joint *P [R_1 ] 30%       FINE
       5:Joint *P [R_2 ] 30%       FINE
    [End]


    Teach Route Points

      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.

```
PROGRAM1
     PAL_1[A_2]UF:0   UT:1      CONF:NUT 000
     X    516.129   mm  W     180.000  deg
     Y    111.347   mm  P       0.000  deg
     Z   1010.224   mm  R       0.000  deg
    Position Detail
    PALLETIZING Route Points
          IF PL[ 1]=[*,*,*]              1/3
       1:Joint  P [A_2 ] 30%      FINE
       2:Joint *P [A_1 ] 30%      FINE
       3:Joint *P [BTM ] 30%      FINE
       4:Joint *P [R_1 ] 30%      FINE

    Enter value
                      CONF     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
    PALLETIZING Route Points

          IF PL[ 1]=[*,*,*]             1/3
       1:Joint  P [A_2 ] 30%      FINE
       2:Joint *P [A_1 ] 30%      FINE
       3:Joint *P [BTM ] 30%      FINE
       4:Joint *P [R_1 ] 30%      FINE
       5:Joint *P [R_2 ] 30%      FINE
    [End]

                                        POSITION
```

5 Pressing [FWD] key while holding down [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 display the next path pattern teaching screen. If the path pattern is only one pattern, go to the step 9.

```
PROGRAM1
     PALLETIZING Route Points

           IF PL[ 1]=[*,*,*]              2/3
        1:      *P [A_2 ]
        2:      *P [A_1 ]
        3:      *P [BTM ]
        4:      *P [R_1 ]
        5:      *P [R_2 ]
     [End]



     Teach Route Points
    BACK               RECORD   DONE    >
```

8 Press F1, BACK to specify the previous path pattern. Press F5, DONE to specify the next path pattern.

9 After teaching of all the path patterns is complete, press F5, DONE to exit from the palletizing edit screen and display the program screen again. The palletizing instruction is automatically written in the program.

```
PROGRAM1
                                      13/13
        5: L  P[2] 300mm/sec CNT50
        6:   PALLETIZING-EX_4
        7: J PAL_4[A_2 ] 30% FINE
        8: J PAL_4[A_1 ] 30% FINE
        9: J PAL_4[BTM ] 30% FINE
       10: J PAL_4[A_1 ] 30% FINE
       11: J PAL_4[A_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.

```
PROGRAM1
                                      14/14
        5: L  P[2] 300mm/sec CNT50
        6:   PALLETIZING-EX_4
        7: J PAL_4[A_2 ] 30% FINE
        8: J PAL_4[A_1 ] 30% FINE
        9: J PAL_4[BTM ] 30% FINE
       10:   hand_open
       11: J PAL_4[A_1 ] 30% FINE
       12: J PAL_4[A_2 ] 30% FINE
       13:   PALLETIZING-END_4
      [End]

   POINT                        TOUCHUP    >
```

For details of palletizing programs, see Section 10.4, “Executing the Palletizing Function”.

10.3.6 Notes on Teaching the Palletizing Function

- The palletizing function is enabled only when a program contains these three instructions: A palletizing instruction, palletizing motion instruction, and palletizing end instruction. When just one of the three instructions is taught into a subprogram by another operation such as copying, normal operation cannot be performed.
- When all palletizing data has been taught, palletizing numbers are automatically written together with the instructions (a palletizing instruction, palletizing motion instruction, and palletizing end instruction). The user need not be concerned about the duplication of these numbers in other programs. (Each program has its own data for palletizing numbers.)
- In the palletizing motion instruction, C (circular motion) and A (circle arc motion) cannot be specified as the motion format.
- When palletizing, with a system with extended axes, there are some special conditions. For a system with extended axes, refer to Section 10.6 ”Palletizing Function with extended axes”.
- When the position data in bottom points and route points are taught, user frame is not used. World frame is always used in teaching these points, and user frame number is the position data of these points is always 0.
- Do not add the time before function to the palletizing motion instruction. Palletizing motion instruction may not function correctly.
- Programs including palletizing instructions cannot use the coordinate system change shift function.

10.4 EXECUTING THE PALLETIZING FUNCTION

Palletizing program

The execution of the palletizing function is shown below.

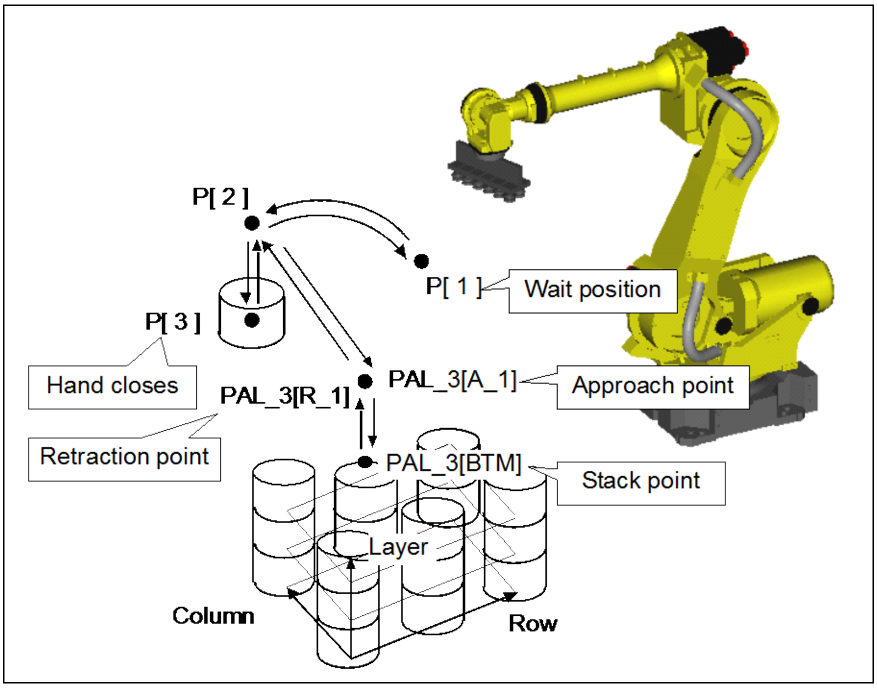


Fig. 10.4 (a) Example of palletizing

Example

```
 5: J  P[1] 100% FINE
 6: J  P[2] 70% CNT50
 7: L  P[3] 50mm/sec FINE
 8:    hand close
 9: L  P[2] 100mm/sec CNT50
10:   PALLETIZING-B_3
11: L PAL_3[ A_1 ] 100mm/sec CNT10
12: L PAL_3[ BTM ] 50mm/sec FINE
13:   hand open
14: L PAL_3[ R_1 ] 100mm/sec CNT10
15:   PALLETIZING-END_3
16: J  P[2] 70% CNT50
17: J  P[1] 100% FINE
```

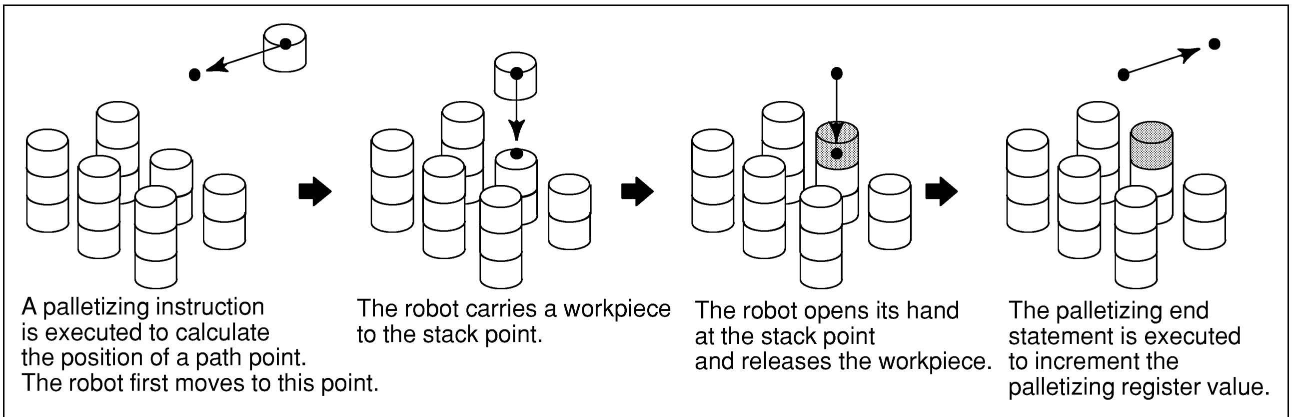


Fig. 10.4 (b) Workpiece stacking process

10.4.1 Palletizing Register

The palletizing register manages the position of the current stack point. When a palletizing instruction is executed, the value held in the palletizing register is referenced, and the actual stack point and path points are calculated. (See Section 7.5, “PALLETIZING REGISTER”.)

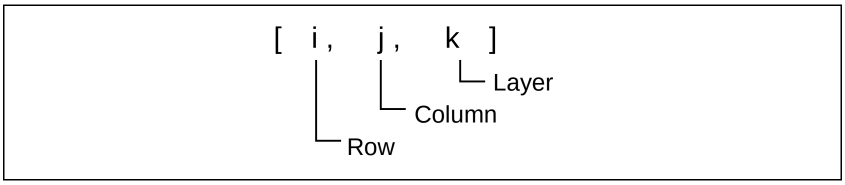


Fig. 10.4.1 (a) Palletizing register

The palletizing register indicates the row, column, and layer which are used for calculating the position of the stack point when the palletizing instruction is executed.

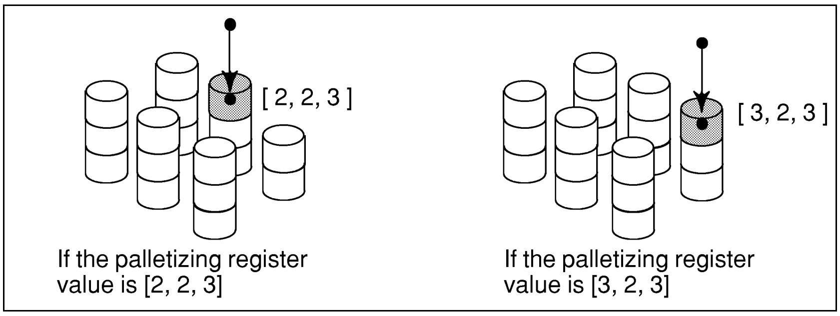


Fig. 10.4.1 (b) Relationship between the palletizing register and stack point

Updating the Palletizing Register

The palletizing register is incremented (or decremented) by executing the palletizing end instruction. The increment (decrement) method is determined depending on the initial data setting.
For 2-row, 2-column, and 2-layer palletizing with ORDER = [RCL] specified, executing the palletizing end instruction changes the palletizing register as follows:

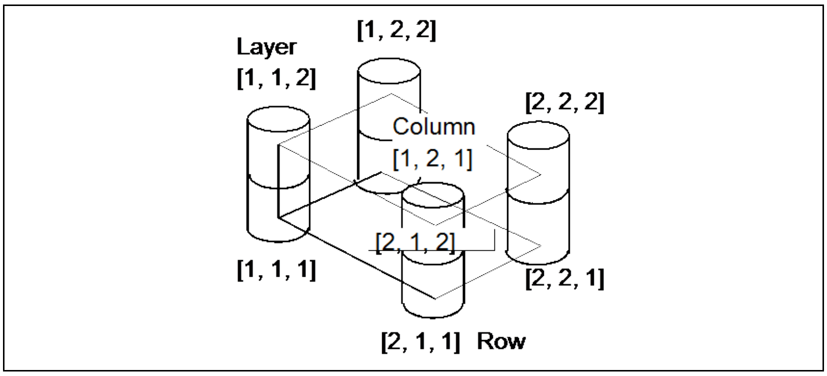


Fig. 10.4.1 (c) Example of 2-row, 2-column, and 2-layer palletizing

Table 10.4.1 (a) Incrementing (Decrementing) order of the palletizing register

	TYPE = [PALLET]		TYPE = [DEPALLET]	
	INCR = [1]	INCR = [-1]	INCR = [1]	INCR = [-1]
Initial value	[1, 1, 1]	[2, 2, 1]	[2, 2, 2]	[1, 1, 2]
↓	[2, 1, 1]	[1, 2, 1]	[1, 2, 2]	[2, 1, 2]
↓	[1, 2, 1]	[2, 1, 1]	[2, 1, 2]	[1, 2, 2]
↓	[2, 2, 1]	[1, 1, 1]	[1, 1, 2]	[2, 2, 2]
↓	[1, 1, 2]	[2, 2, 2]	[2, 2, 1]	[1, 1, 1]
↓	[2, 1, 2]	[1, 2, 2]	[1, 2, 1]	[2, 1, 1]
↓	[1, 2, 2]	[2, 1, 2]	[2, 1, 1]	[1, 2, 1]
↓	[2, 2, 2]	[1, 1, 2]	[1, 1, 1]	[2, 2, 1]
↓	[1, 1, 1]	[2, 2, 1]	[2, 2, 2]	[1, 1, 2]

Initializing the palletizing register

When F5, DONE is pressed upon completion of the setting or changing of initial palletizing data, the palletizing stacking pattern teach mode is set. At this time, the palletizing register is initialized automatically. (See Subsection 10.3.2, “Inputting Initial Data”.)

Table 10.4.1 (b) Initial value of the palletizing register

Initial data		Initial value		
TYPE	INCR	ROWS	COLUMNS	LAYERS
PALLET	Positive value	1	1	1
	Negative value	Total number of rows	Total number of columns	1
DEPALLET	Positive value	Total number of rows	Total number of columns	Total number of layers
	Negative value	1	1	Total number of layers

10.4.2 Controlling the Palletizing Function by a Palletizing Register

In 5-row, 1-column, and 5-layer palletizing, suppress stack operation for the fifth workpiece in each even-numbered layer. (Stack five workpieces in odd-numbered layers, and stack four workpieces in even-numbered layers.)

```
PROGRAM2
                                          1/17
      1:   PL[1]=[1,1,1]
      2:   LBL[1]
      3:   IF PL[1]=[5,*,2-0]
            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

```
PROGRAM2
                                          6/17
      4: L  P[1] 100mm/sec FINE
      5:   hand_close
      6:   PALLETIZING-B_1
      7: L PAL_1[A_1  ] 100mm/sec CNT30


    Select item

   [MODIFY]                [CHOICE]   LIST
```

Step

1 To display the palletizing status, place the cursor on the palletizing instruction, then press F5, LIST. The current stack point and the value of the palletizing register are displayed.

```
PROGRAM2
   PALLETIZING-B_1   [                   ]
    CURRENT BOTTOM POINT [  3,  1,  1]
    PALLET REGISTER [ 1]=[  4,  1,  1]
    ROUTE PATTERN[ 1]=[*,*,*]
   Pallet List
```

10.5 MODIFYING THE PALLETIZING FUNCTION

The palletizing data and palletizing instructions which were taught can be modified later.

Procedure 10-7 Modifying palletizing data

```
PROGRAM2
                                        6/17
        4: L  P[1] 100mm/sec FINE
        5:   hand_close
        6:   PALLETIZING-B_1
        7: L PAL_1[A_1  ] 100mm/sec CNT30


     Select item
   [MODIFY]                  [CHOICE]  LIST
```

Step

1 Place the cursor on the palletizing instruction which is to be modified, then press F1, [MODIFY] to display the modification menu.

```
PROGRAM2
                                        6/17
        4: L  P[1] 100mm/sec FINE
        5:   hand_close
        6:   PALLETIZING-B_1
        7: L PAL_1[A_1  ] 100mm/sec CNT30

     MODIFY  1
 1 CONFIG
 2 BOTTOM
 3 PATTERN
 4 ROUTE
   [MODIFY]                  [CHOICE]  LIST
```

2 Select a target palletizing edit screen in the modification menu.

```
PROGRAM2
    PALLETIZING Bottom Points

       1:--P [  1,  1,  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.

Procedure 10-8 Changing the palletizing number

```
PROGRAM2
                                         6/17
     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  ] 100mm/sec CNT30
    11:   PALLETIZING-END_1

   Select item
 [MODIFY]              [CHOICE]  LIST
```

Step

1 Place the cursor on the palletizing instruction having the palletizing number which is to be changed, then enter a new number.

```
PROGRAM2
                                         6/17
     6:   PALLETIZING-B_2
     7: L PAL_2[A_1  ] 100mm/sec CNT30
     8: L PAL_2[BTM  ] 50mm/sec FINE
     9:   hand_open
    10: L PAL_2[R_1  ] 100mm/sec CNT30
    11:   PALLETIZING-END_2

   Select item
 [MODIFY]              [CHOICE]  LIST
```

At the same time the palletizing number of the palletizing instruction is changed, the palletizing numbers of the palletizing motion and palletizing end instructions are also changed.

> ⚠ **CAUTION**
> When changing palletizing numbers, make sure that the new numbers are not used by other palletizing instructions.

10.6 PALLETIZING FUNCTION WITH EXTENDED AXES

This section explains the special items when using palletizing on a system with extended axes.
When teaching the bottom point or route points for palletizing, which are different from the usual teaching of motion instruction, the position which is removed the position of extended axes is recorded.

Execution

When palletizing is executed, palletizing will be done at the position of the extended axes at that time. (The robot does not automatically return to the position of the extended axes at teaching bottom points or route points.) For example, on a system with a linear extended axis, when the palletizing instruction is executed at a point 1000 mm away from the position at teaching palletizing, the robot will perform the same motion as taught at the position which is 1000 mm away from the teaching position.

In case that the skip instruction is taught in the motion instruction immediately before the palletizing instruction, when the skip condition is satisfied during the execution of that motion instruction, the palletizing is executed at the position of the extended axis when the skip condition is satisfied.

Position teaching/modifying

When teaching the bottom/route points for palletizing on a system with extended axes or modifying the position, the following attention is necessary.

- When teaching bottom/route points for palletizing, it must be done at the fixed point.
  - When the position of extended axes shifts while teaching the bottom/route points for the same palletizing, the robot will follow a different path 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.

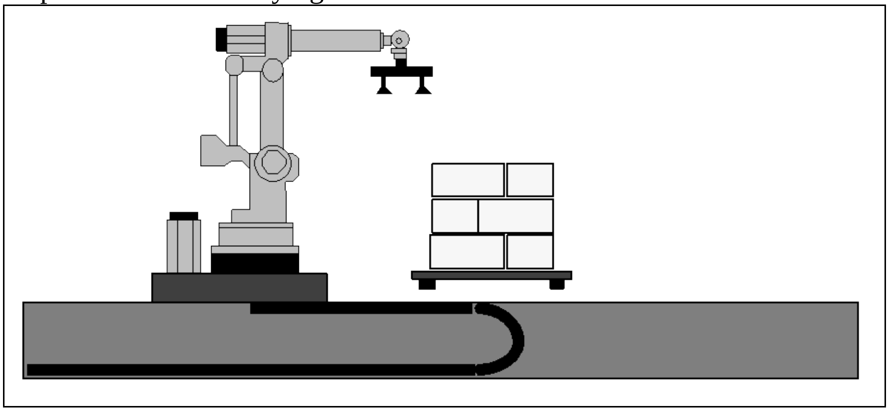


Fig. 10.6 Palletizing function with extended axes

10.7 PALLETIZING ALL-POINT TEACHING

Palletizing all-point teaching allows the stacking (or unstacking) of workpieces without changing the taught configuration by changing a system variable.

Operation

Set the following values to stack (or unstack) workpieces without changing the taught posture and configuration:

1. On the system variable screen, set system variable $PALCFG.$FREE_CFG_EN to TRUE (the initial value is TRUE).
2. On the palletizing initial data screen, set INTER for posture control in the row, column, or layer direction for which FREE is specified as the arrangement mode.

   For each taught workpiece in the specified direction, all workpieces corresponding to the taught workpiece are stacked (or unloaded) with the same posture and configuration as that for the taught workpiece.

The following shows an example of palletizing for an irregular arrangement of four rows, two columns, and five layers.

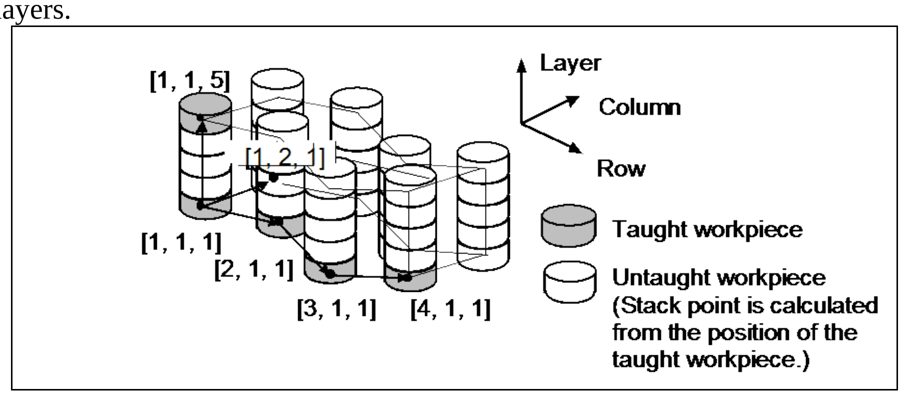


Fig. 10.7 Palletizing all-point teaching

The following palletizing initial data is set:

```
ROWS=       [4   FREE   INTER]
COLUMNS  =  [2   LINE   FIX  ]
LAYERS   =  [5   LINE   FIX  1]
```

In this example, FREE and INTER are set in the row direction. When system variable $PALCFG.$FREE_CFG_EN is set to TRUE under this condition, the following configurations are used:

- Configuration for stacking (unloading) workpieces on row 1: Configuration in P[1,1,1]
- Configuration for stacking (unloading) workpieces on row 2: Configuration in P[2,1,1]
- Configuration for stacking (unloading) workpieces on row 3: Configuration in P[3,1,1]
- Configuration for stacking (unloading) workpieces on row 4: Configuration in P[4,1,1]

Notes

Note the following points when using this function:

1 FREE and INTER can be set at the same time in only one of the row, column, and layer directions (when this function is not to be used, set system variable $PALCFG.$FREE_CFG_EN to FALSE).

This is because if FREE and INTER are set at the same time in two or more of the row, column, and layer directions, two or more configurations to be taken at the position of an untaught workpiece (workpiece for which the stack point is calculated from the position of a taught workpiece) are made.

If such a setting is made in a program, the program causes the error indicated by PALT-024 Calculation error occurred and cannot be executed.

2 Carefully teach a program so that the program is not stopped due to a configuration mismatch alarm.

If the configuration in the current position differs from the configuration data for the destination position, the robot cannot move in Linear operation mode (a configuration mismatch alarm occurs and execution of the program is stopped).

The configuration at a stack point is used for the configuration at an approach or retraction point during palletizing. Therefore, if the pallet operation instruction to be executed first is in Linear operation mode, a configuration mismatch may occur depending on the configuration of the robot when an attempt is made to execute the line.

To avoid such a problem, specify Joint for the operation mode of the first pallet operation instruction.

For example, to avoid a configuration mismatch alarm, the following programming can be used for palletizing with three approach points and two retraction points.

```
  :
10:   PALLETIZING-EX_1
11: J PAL_1[A_3 ] 100% FINE
12: L PAL_1[A_2 ] 500mm/sec CNT50
13: L PAL_1[A_1 ] 300mm/sec CNT10
14: L PAL_1[BTM ] 100mm/sec FINE
15:   Open hand 1
16: L PAL_1[R_1 ] 300mm/sec CNT10
17: L PAL_1[R_2 ] 500mm/sec CNT50
18:   PALLETIZING-END_1
  :
```

11 APPLICATION FUNCTION OF TEACH PENDANT

This chapter describes application function of teach pendant.
Contents of this chapter

11.1 SCREEN DISPLAY FUNCTIONS

11.1.1 Internet Browser Screen

To display the Internet browser screen, press [MENU] key. The following screen menu will be displayed:

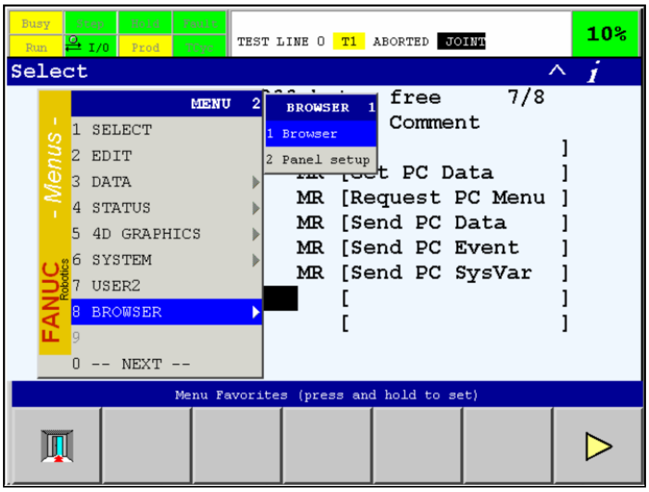


Fig. 11.1.1 (a) Screen menu

Select “BROWSER” from the screen menu. The following screen will be displayed:

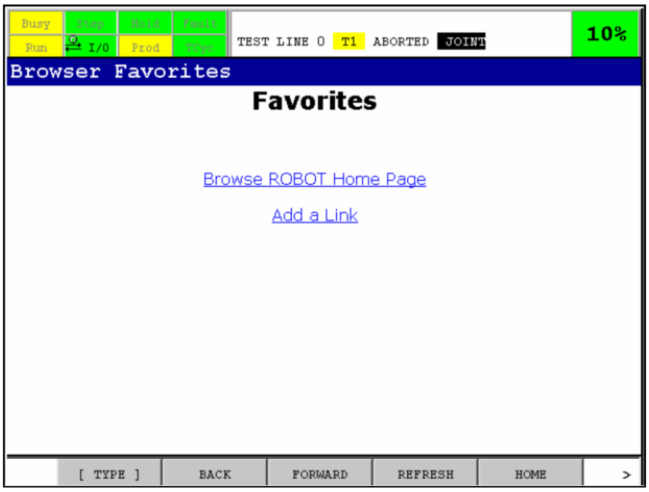


Fig. 11.1.1 (b) Internet browser screen

Select "Add a Link" and press the [ENTER] key. The following URL input screen will be displayed :

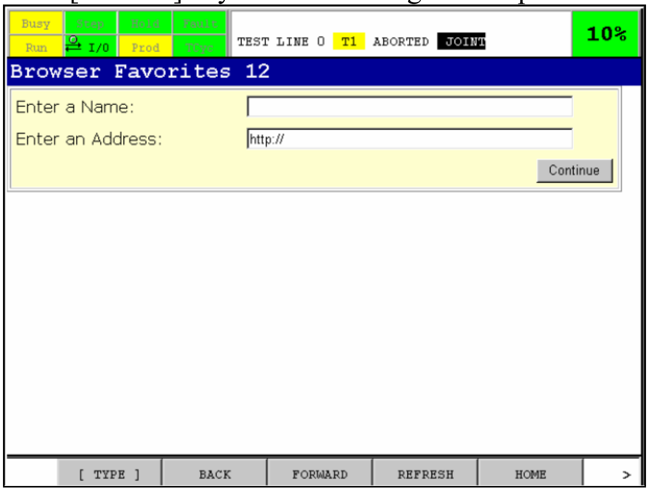
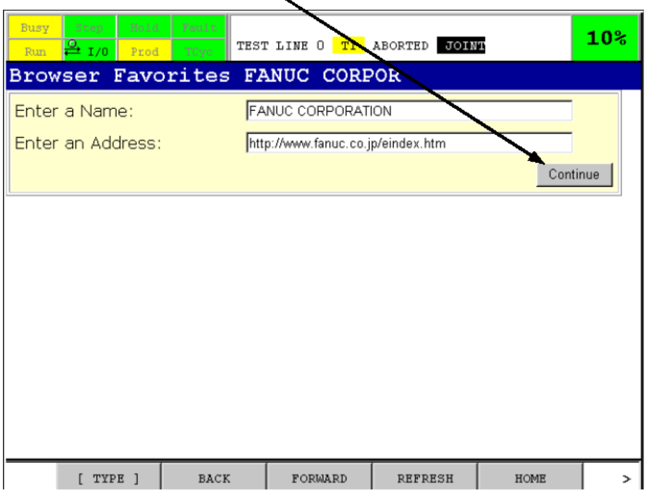


Fig. 11.1.1 (c) URL input screen

Position the cursor on "Enter a Name" or" Enter an Address" in the above screen and press the [ENTER] key.
The following software keyboard will be displayed . Enter alphabetic and other characters.
After confirming your entry, press the <u>exit button</u> at the upper right to exit the software keyboard.

In the following sample screen, the name and address have been entered. After you have entered the name and address, position the cursor on the <u>Continue button</u> and press the [ENTER] key.



After the Continue button is pressed, the registered link information is displayed as follows:

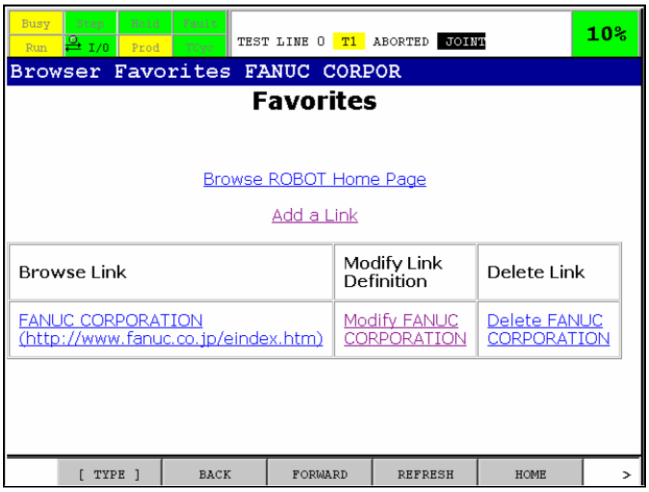


Fig. 11.1.1 (d) Link screen

> **NOTE**
> Links definition is saved to CUSTLIST.DT when you backup using "All of above" from file screen. Zero size CUSTLIST.DT will be saved when you backup if you remove all links. And FILE-062 alarm can be occurred. Add any link if FILE-062 alarm is occurred.

11.1.2 Status Sub-window

The status sub-window displays various types of statuses graphically. To display the status sub-window, select “4 Status/Single” from the following screen menu:

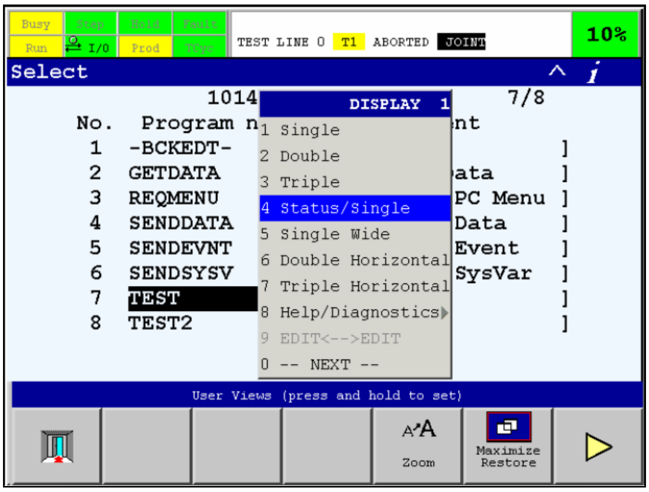


Fig. 11.1.2(a) Screen switch menu

The left screen of the following two screens is the status sub-window.
Position the cursor on Position Display, Operator Panel or Stop Signals and press the [ENTER] key. The corresponding status screen will be displayed.

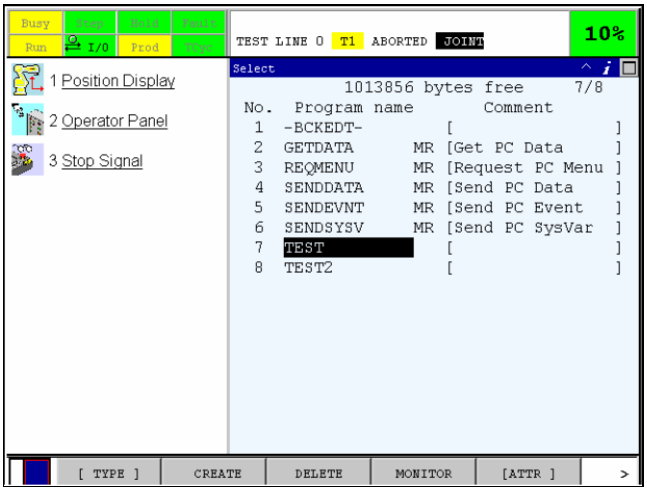


Fig. 11.1.2(b) Status subwindow

11.1.2.1 Current position display

When "Position Display" is selected, the following screen will be displayed:

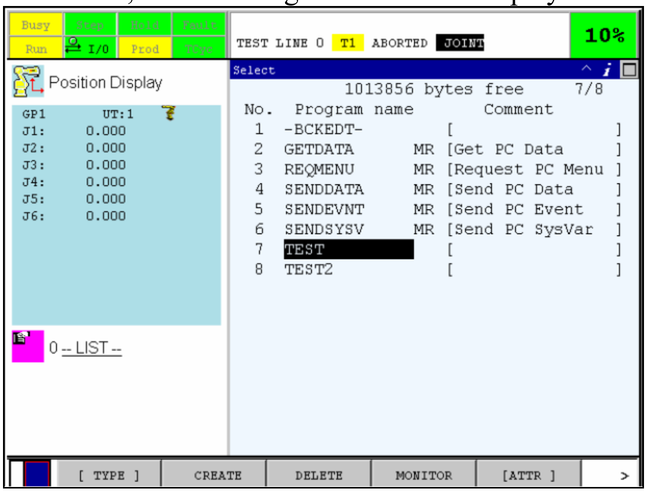


Fig. 11.1.2.1 Status subwindow (Position display)

11.1.2.2 Operator panel status display

When "Operator Panel" is selected, the following screen will be displayed. Each graphic indicator is on or off according to the status of the remote device.

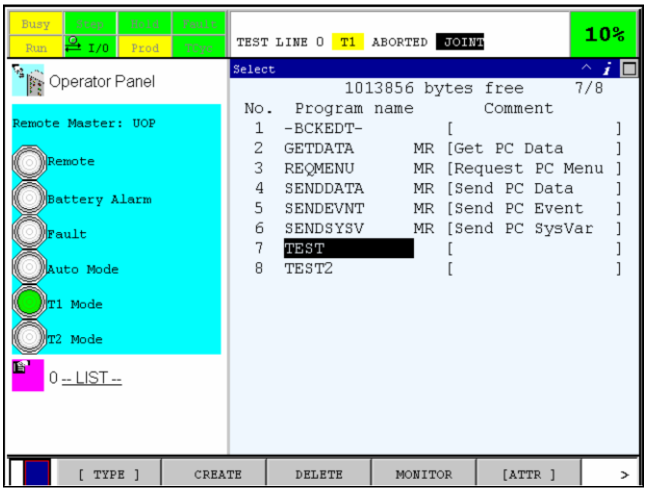


Fig. 11.1.2.2 Status subwindow (Operator panel)

11.1.2.3 Stop signal status display

When “Stop Panel” is selected, the following screen will be displayed. Each graphic indicator is on or off according to the status of the corresponding stop signal.

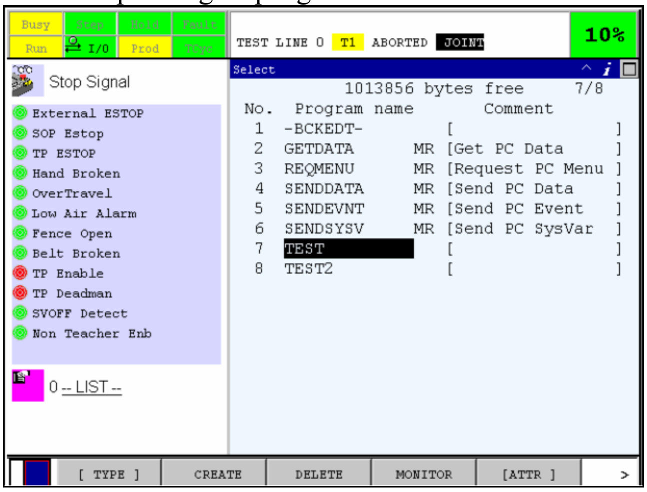


Fig. 11.1.2.3 Status subwindow (Stop signal)

11.1.3 Wide Screen Mode

In DISP menu, by selecting “Single Wide”, until 76 characters per line and 20 lines can be displayed. This is useful to display long line screens and many line screens like program edit screen and alarm history etc.

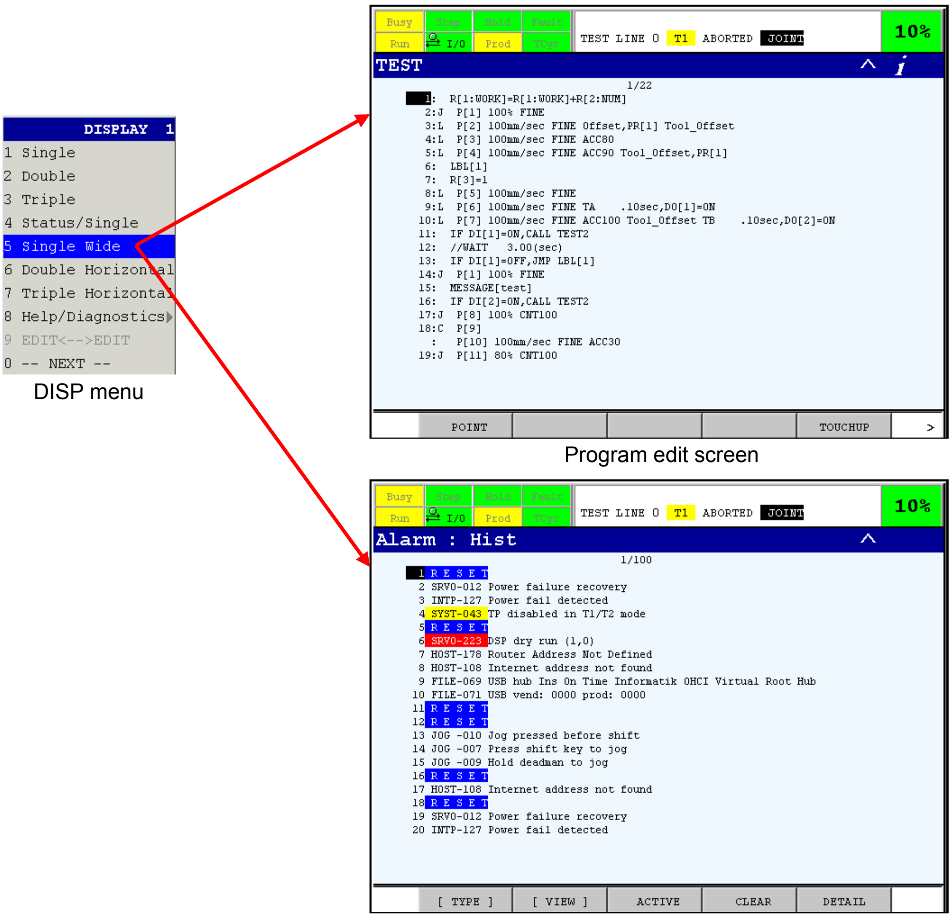


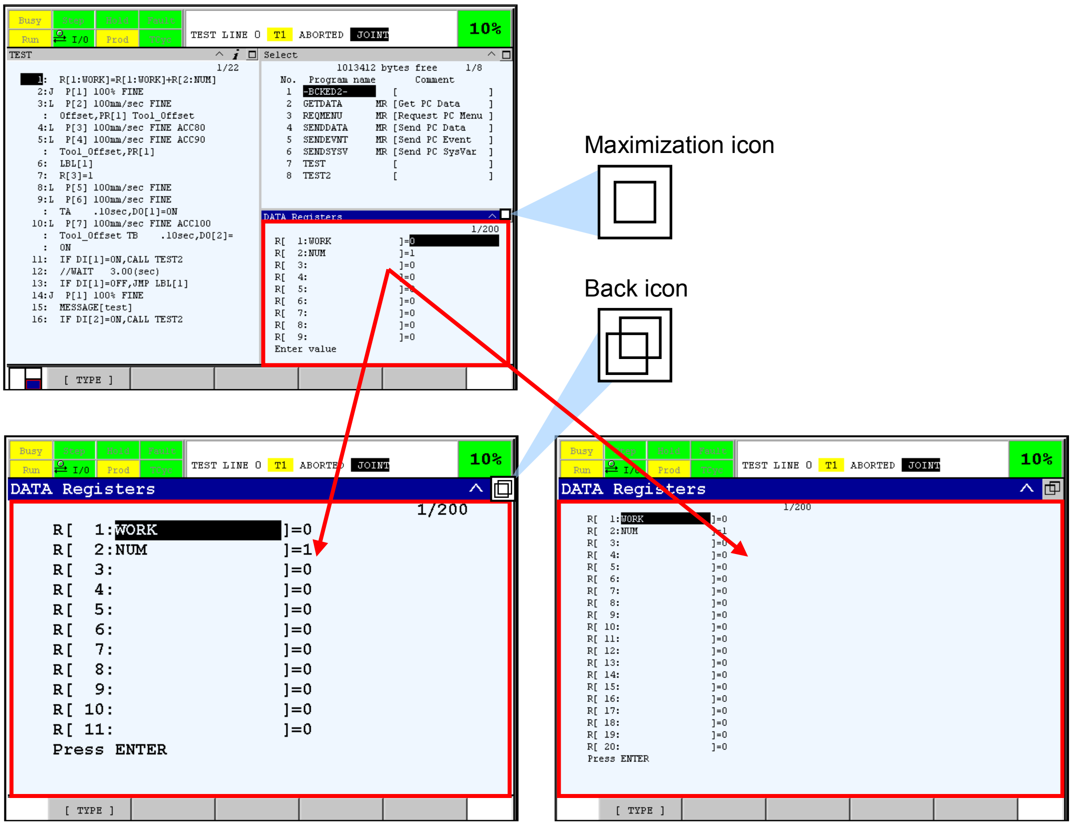
Fig. 11.1.3 Wide screen

11.1.4 Screen Maximization

In double and triple window mode, it might be possible to feel that the screen is not easy to look at because that character font is too small. The screen maximization function can be used to change the current screen to full screen mode temporarily and return to the original screen mode.

There are some methods to maximize and back to original size.

1. Press the maximization icon or the back icon on the upper right of the window by a touch panel.
2. Press the maximization icon and the back icon in ICON menu if these icons are not deleted from your ICON menu.
3. Press “SHIFT” key + DISP key to show DISP menu -> select “0:NEXT” -> select “Maximize/Restore”.



Maximized screen (Full screen mode) Maximized screen (Wide screen mode)

NOTE

1. Maximize operation can work in double, triple and single with status screen mode. If current screen mode is single or wide mode and the maximize operation is done but the screen will not be changed.
2. The maximized screen is displayed in the recent of which the single mode or the wide mode.

11.1.5 Zoom

It might be possible to feel that the screen is not easy to look at because that character font is small. The zoom function can be used to change the current character font size to big temporarily and return to the original size.

The character font size is enlarged by the zoom operation. Furthermore, if the zoom operation is executed again, the character font size becomes one size larger. Then if the zoom operation executed again, the character font size is returned to the original size. (In single screen, the character font size is enlarged only once.) If the screen width exceeds the window width, the lateral scroll bar appears at the bottom of the screen. If the teach pendant equips the touch panel, the screen can be scrolled in lateral direction by touching the scroll bar. If the teach pendant does not equip the touch panel, the screen can be scrolled in lateral direction by pressing right or left arrow key with [SHIFT] key.

There are some methods to zoom and back to original size.

1. Press the zoom icon or the back icon on the upper right of the window by a touch panel.
2. Press the zoom icon and the back icon in ICON menu if these icons are not deleted from your ICON menu.
3. Press “SHIFT” key + DISP key to show DISP menu -> select “0:NEXT” -> select “Zoom”.

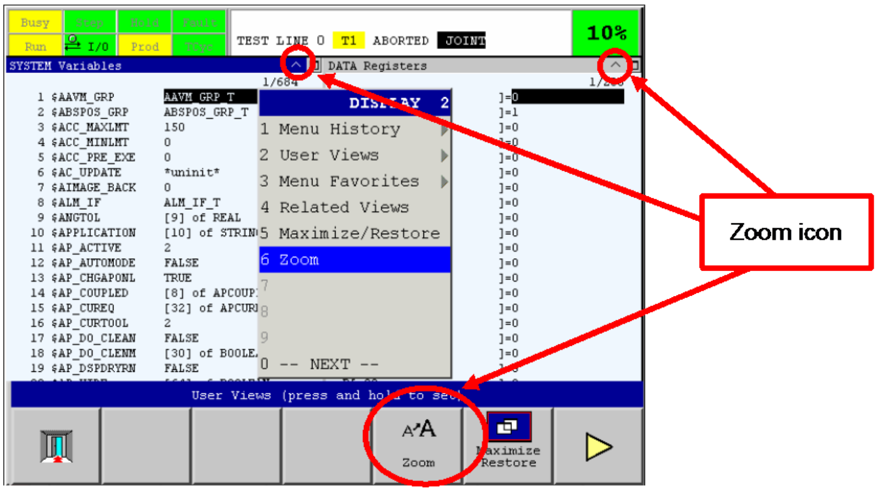
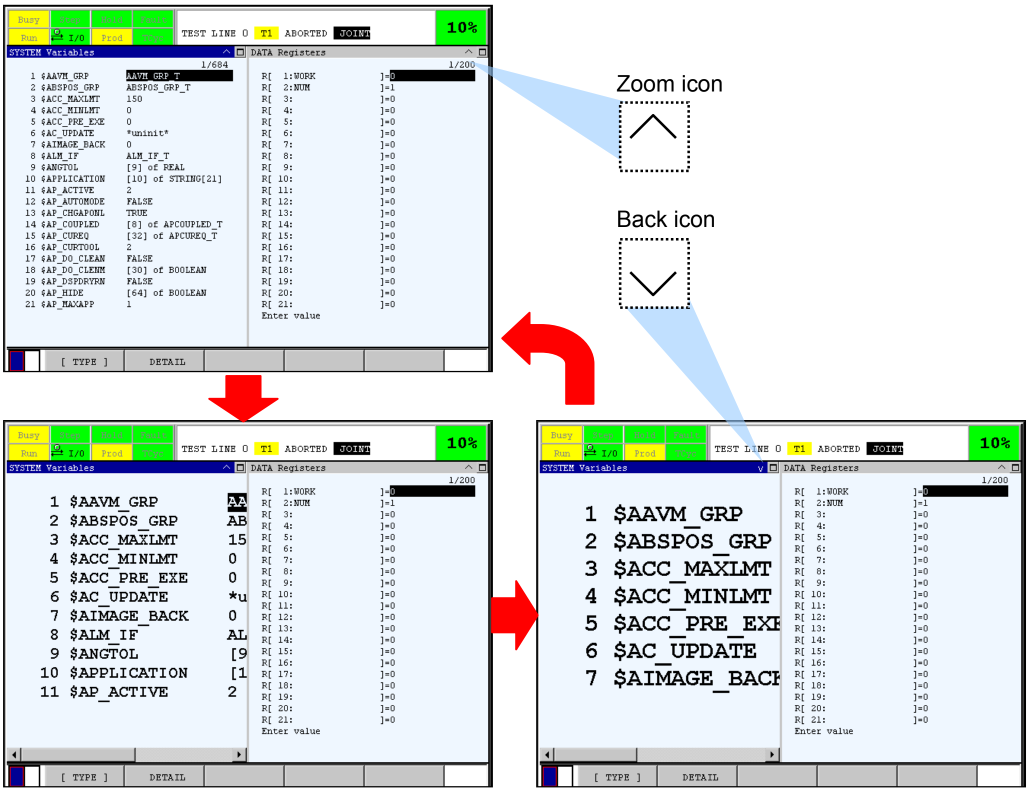


Fig. 11.1.5(a) Zoom icon



Zoomed screen (Left side) The screen zoomed again (left side)

Fig. 11.1.5(b) Zoomed screen

> **NOTE**
> If move the other screen or change the window mode, return the default font size.

11.1.6 Tree View

It is possible to display various items, which were conventionally displayed as list type, as tree view type. The screen is divided to two screens as Fig. 11.1.6(a). And Tree View is on the left side, main screen is on the right side, and also the main screen is changed with operation on Tree View. To display Tree View, there are two ways, the one way is to select Tree View on Screen menu (which can be displayed by “SHIFT” key and DISP key simultaneously) as Fig. 11.1.6(b), the other way is to select an item on Top Menu/Tree View as Fig. 11.1.6(c).

> **NOTE**
> Tree View function is available on software version 7DC3 series or later.

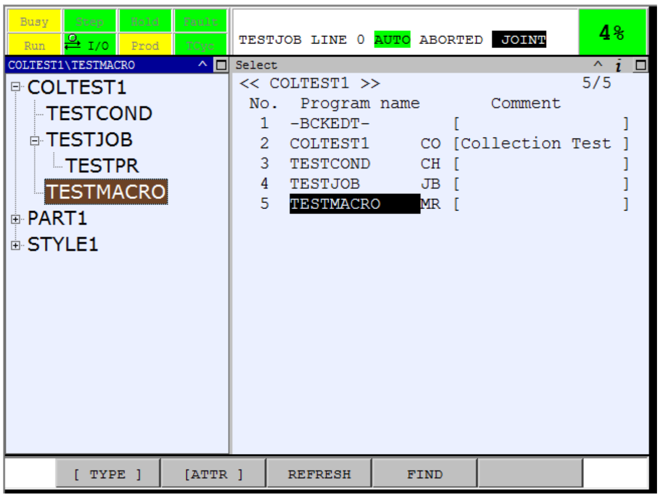


Fig. 11.1.6(a) Tree View Screen

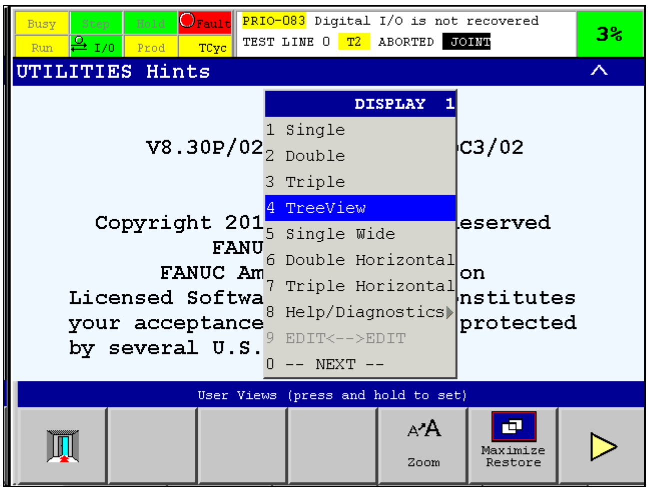


Fig. 11.1.6(b) Screen Menu

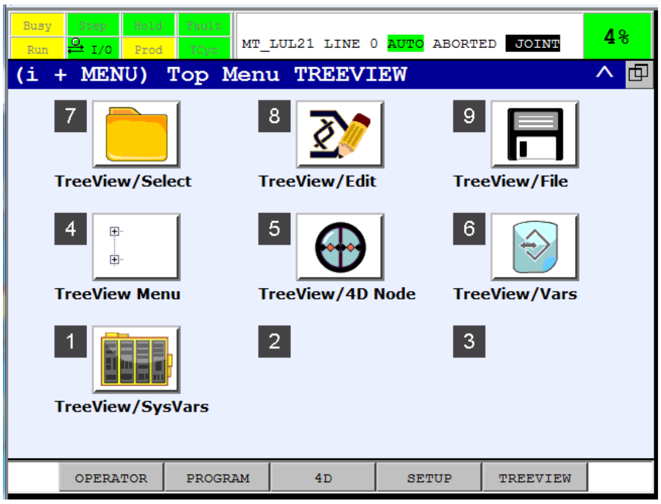


Fig.11.1.6(c) Top Menu/Tree View

There are the following types of Tree View as Table11.1.6(a).

Table 11.1.6 (a) Tree View types

TYPE	DESCRIPTION
All programs	Displays all Robot application programs on Tree View.
Call Trees	Displays Robot application programs having sub programs and the sub programs on Tree View.
Collections	Displays Collections and programs registered to the collection programs on Tree View.
Menus	Displays Screen Menus on Tree View.
Files	Displays File I/O devices. Each device opens to display its files and subdirectories on Tree View.
System Variables	Displays System variables on Tree View.
Jobs	Displays Jobs (Programs having subtype JOB) Each job opens to display its called programs on Tree View.

There are the operations of Tree View as Table11.1.6(b).

Table 11.1.6 (b) Operations of Tree View

KEY	DESCRIPTION
F1[TYPE] key	Use this function key to display different types.
F2[ATTR] key	Use this function key for program types and display and sort the tree with Name Only, Name/Comment, and Comment/Name.
F3[REFRESH] key	Use this function key to refresh the contents of Tree View.
F4[FIND] key	Use this function key to search the entire tree for entered string.
[ENTER] key	Select the TreeView item that is currently highlighted. And expand or collapse the item.
Right arrow key	When key cursor is on a item having sub item(s), the item is expanded. The key cursor is moved to its subitem when the item has been expanded already.
Left arrow key	When key cursor is on a item having sub item(s), the item is collapsed. The key cursor is moved to its main item when its item has been collapsed already.
Up/Down key	Moves key cursor to Up or Down
SHIFT + arrow key	Operates scroll bars.
i + R/L arrow key	When the type is Menus, use these 2 keys together to collapse or expand all menu entries.

11.1.6.1 Tree View/All program types

It is possible to display all robot application programs with Tree View. They are separated to each program subtype items, and it is possible to select a program by selecting the objective program on Tree View. The menu screen displays EDIT screen of the selected program when the main screen is EDIT screen. The main screen displays the Select screen and the key cursor is moved to the selected program when the main screen is the other of EDIT screen.

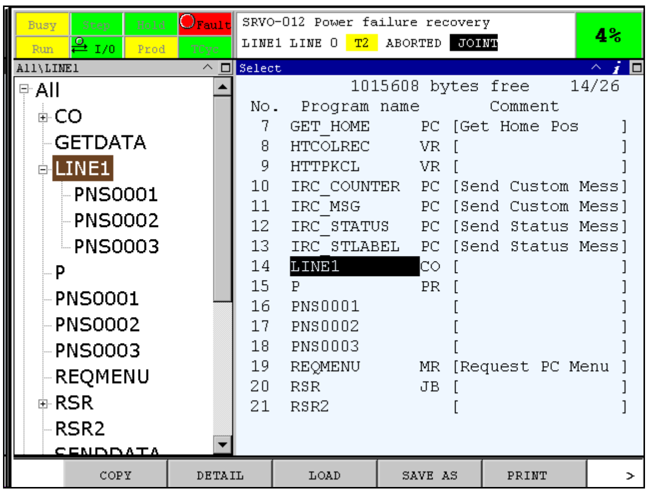


Fig.11.1.6.1 Tree View/Select

11.1.6.2 Tree View/Call Trees

It is possible to display robot application programs having sub programs and the sub programs with Tree View. They are separated to each main program's item, and it is possible to select a program by selecting the objective program on Tree View. The main screen displays EDIT screen of the selected program when the main screen is EDIT screen. The main screen displays the Select screen and the key cursor is moved to the selected program when main screen is the other of EDIT screen.

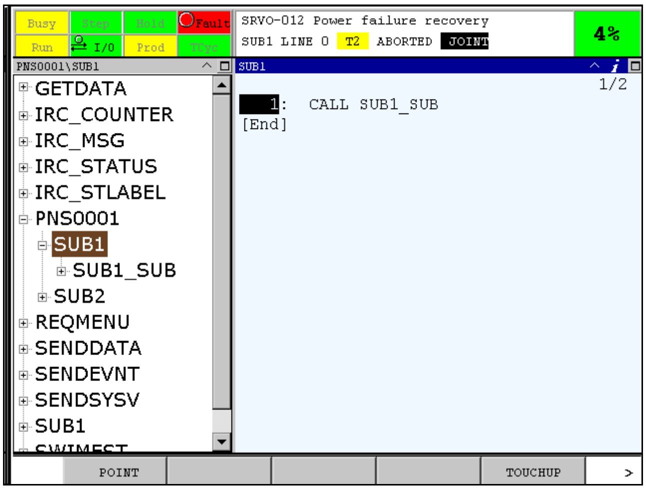


Fig11.1.6.2 Tree View/Call Trees

11.1.6.3 Tree View/Collections

It is possible to display collections and programs registered to the collections with Tree View. They are separated to each collection program item, and it is possible to select a collection or a program by selecting the objective collection or program on Tree View. The menu screen displays EDIT screen of the selected program when the main screen is EDIT screen. The main screen displays the Select screen and the key cursor is moved to the selected program when the main screen is the other of EDIT screen. (Please refer to "5.8.4 Collection program function" in regard to collection program)

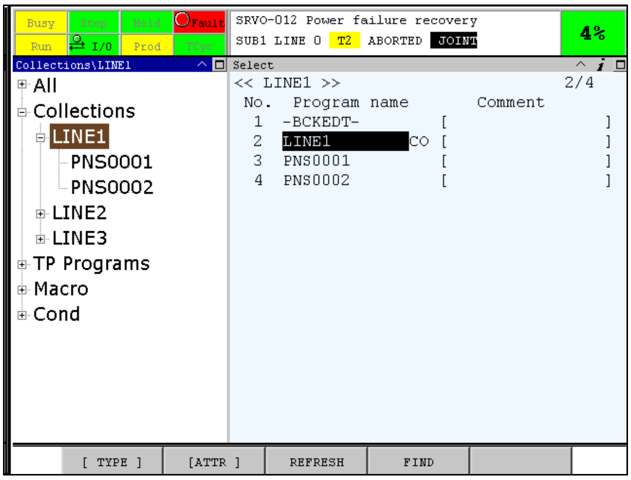


Fig.11.1.6.3 Tree View/Collections

11.1.6.4 Tree View/Menus

Tree View displays Screen Menus. The main menu displays the menu selected on Tree View.

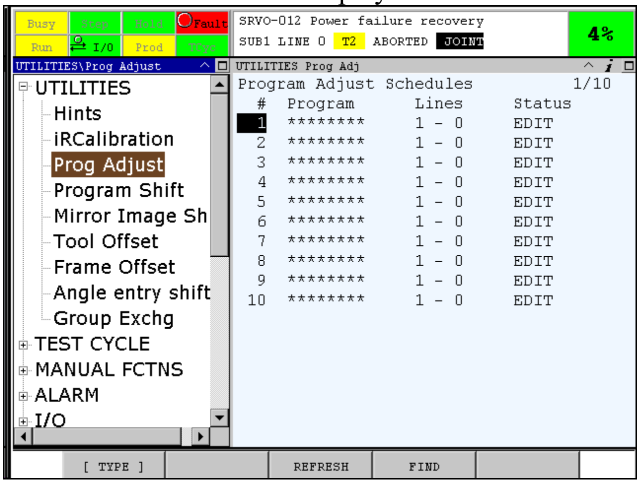


Fig. 11.1.6.4 Tree View/Menus

11.1.6.5 Tree View/Files

Tree View displays File I/O devices and files. They are separated to each File I/O device and sub directory. And also, the main screen becomes file menu by selecting a file on Tree View and key cursor is moved to the selected file.

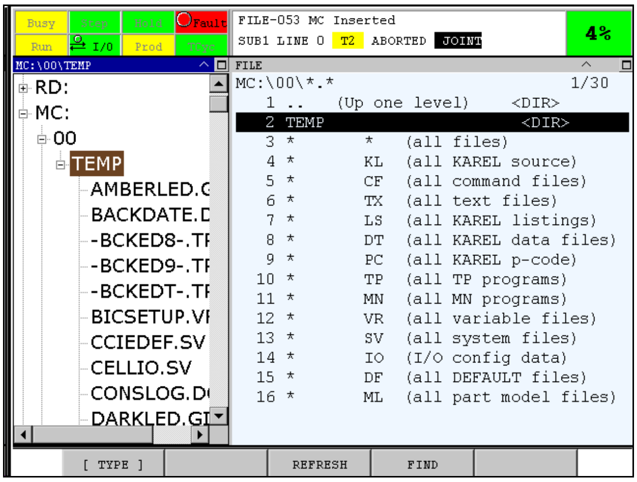


Fig. 11.1.6.5 Tree View/Files

11.1.6.6 Tree View/System Variables

Tree View displays System Variables. They are separated to each initial of system variable name, structure. And also, the main screen becomes system variable screen by selecting a system variable and key cursor is moved to the selected system variable.

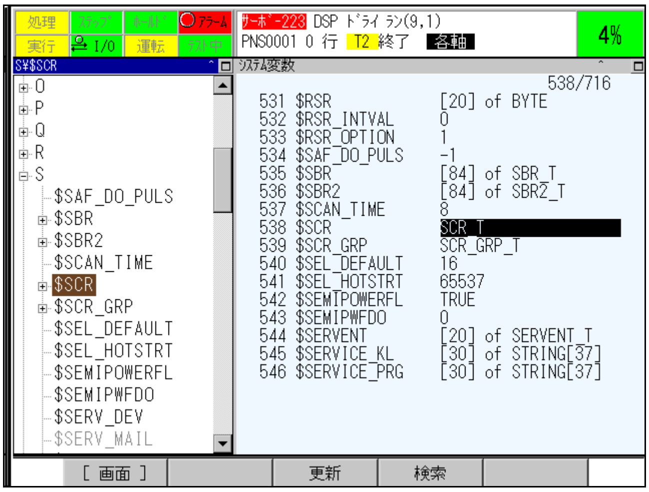


Fig. 11.1.6.6 Tree View/System Variables

11.1.6.7 Searching Tree View

It is possible to search Tree View and to search for a string on Tree View. For instance, when there are a number of robot application programs, you can find out the objective program immediately by using this searching. On the following "Procedure 11-1 Searching Tree View", the procedure, which searches for a string on Tree View, is written.

Procedure 11-1 Searching Tree View

Condition

- The display mode is Tree View.

Step

1 Make Tree View screen be active (The operation on left side screen is active).
2 Press F4[FIND] key so that the string inputting window is displayed as the following figure .

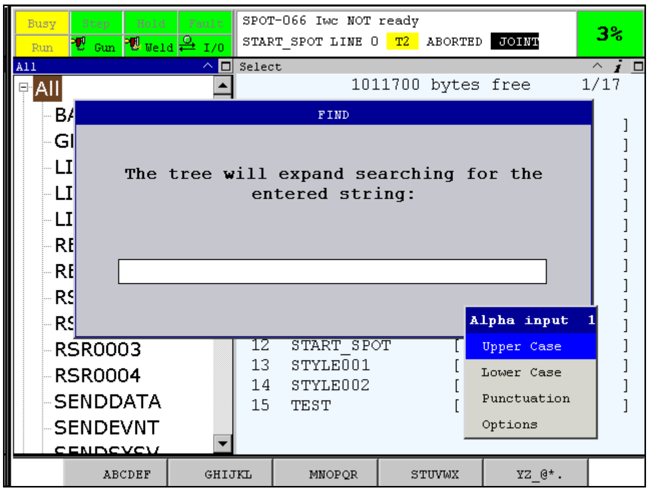


Fig. 11.1.6.7 (a) Searching Tree View 1

3 Input string that you want to search for. (On the following figure, searching for “SPOT”)

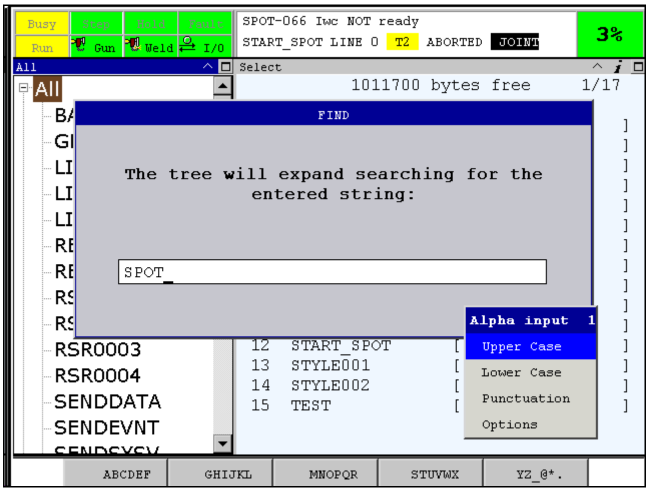


Fig. 11.1.6.7 (b) Searching Tree View 2

4 Press F4[NEXSRCH] key if the displayed item isn’t the target.

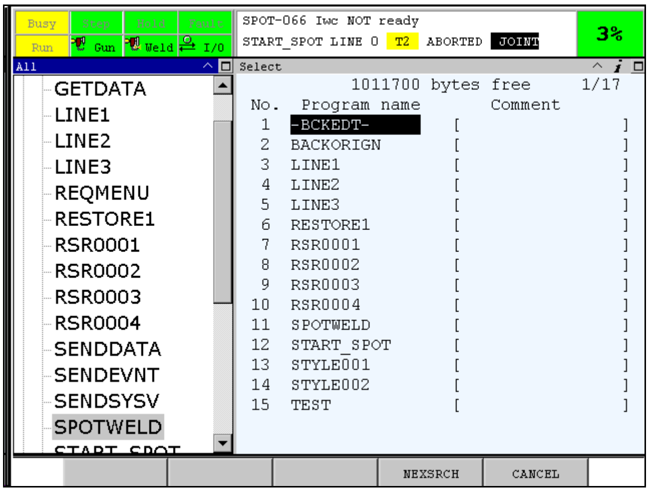


Fig. 11.1.6.7(c) Searching Tree View 3

5 Press the [ENTER] key so that you select the item if the displayed item is the target. Press F5[CANCEL] key or the [PREV] key if you want to cancel searching.

11.2 OPERATION FUNCTIONS

11.2.1 Cursor Positioning by Touch Panel

In the teach pendant which equips the touch panel, a cursor can be moved by touch on the teach pendant screen. Menu invoking, menu item selection, and active window selection also can be done by screen touch.

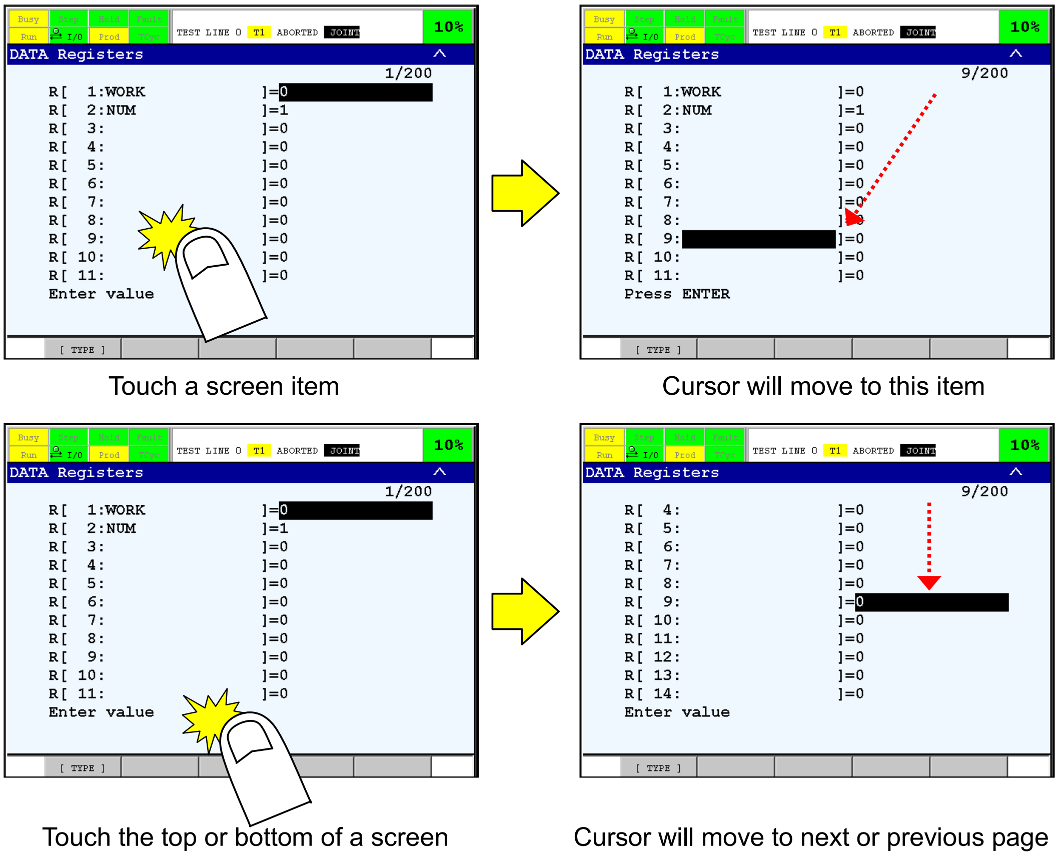
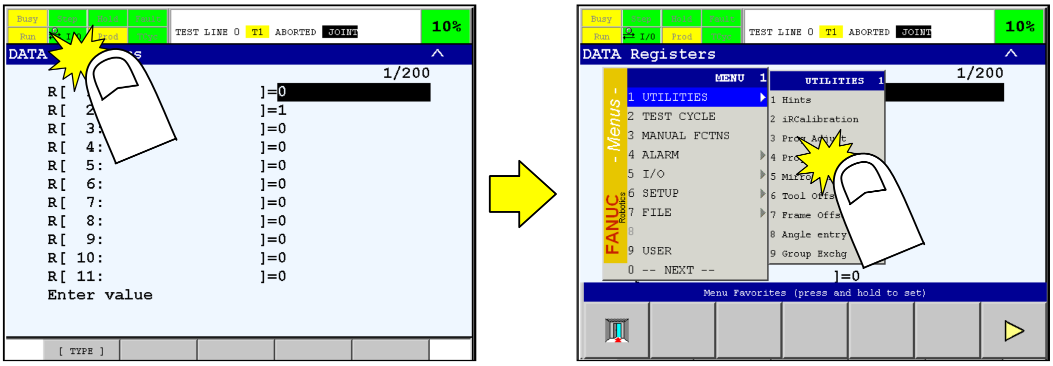
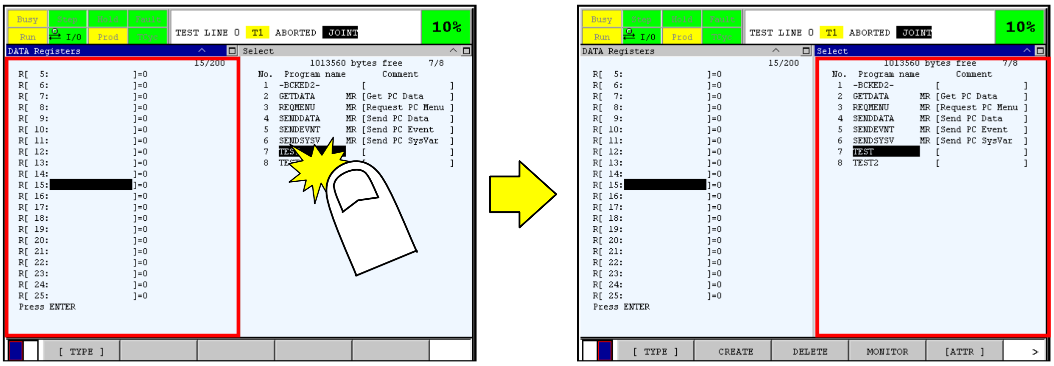


Fig. 11.2.1(a) Cursor positioning by screen touch (1)



Touch the left part of a title bar : Open MENU menu.
Touch the center of a title bar : Open DISP menu.
Touch the right part of a title bar : Open FCTN menu.

Item of a menu can be selected by touch.



Touch the window.

Active window will be changed to the touched side.

Fig. 11.2.1(b) Cursor positioning by screen touch(2)

> **NOTE**
> Touch by a tip of a pen might break the touch panel. Use your finger or the touch pen designed for touch panel operation.

11.3 PROGRAM EDIT FUNCTIONS

11.3.1 ICON Editor

Instead of the traditional program editor of robot controller using pop-up menus to edit instructions of a robot program, it is possible to edit the robot program by touching ICON on the screen.

> **NOTE**
> ICON editor can be used on only the teach pendant which equips the touch panel.

As the default setting, previous traditional editor is available and new ICON editor is disabled. To use new ICON editor, please enable it by the following two procedures.

First:
1. Press [MENU] key while *i* key is pressed.
2. Press F2, PROGRAM key to display Top menu for program.
3. Move the cursor on the item ICON Editor, and press the [ENTER] key, or press 9 in the numeric key.

In order to return the previous traditional editor, push EXIT button in Fig. 11.3.1(a), push F4, EXIT ICON key in F5, EDCMD screen, or select 8 Main Editor in above step 3.

Second:
1 Press the [EDIT] key.
2 Press the [NEXT] key, and press F5, EDCMD key.
3 Move the cursor on the item Icon Editor, and press the [ENTER] key.

In order to return the previous traditional editor, push EXIT button in Fig. 11.3.1(a), push F4, EXIT ICON key in F5, EDCMD screen, or select 8 Main Editor in Top menu.

NOTE
The F4, EXIT ICON key in F5, EDCMD screen is supported from the software version 7DC1(V8.10)/15, the 7DD0(V8.13)/08, the 7DC2(V8.20)/01 or later series.

In ICON editor, the screen on the teach pendant is split to two windows. At the left window, editing program contents are displayed. At the right window, operational ICONs are displayed. The operational ICONs can be selected by direct finger touch on the touch panel. Therefore the operation from touch panel is more intuitive and more efficient from the previous operations in pop-up menus. ICON editor only supports this window configuration.

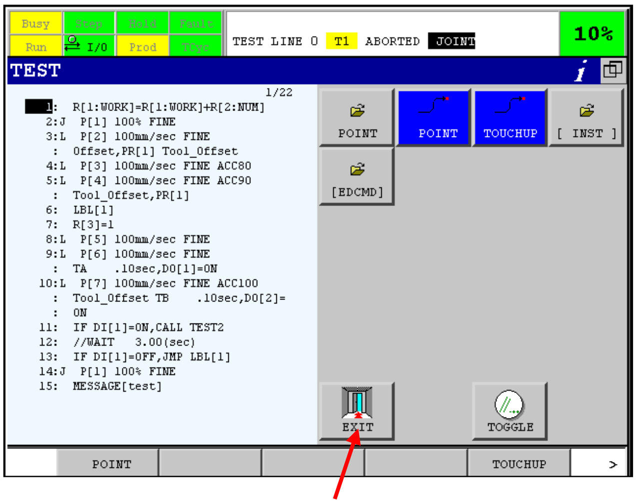


Fig. 11.3.1(a) ICON editor

ICONs in the right window are the same items displayed in the previous pop-up menu.

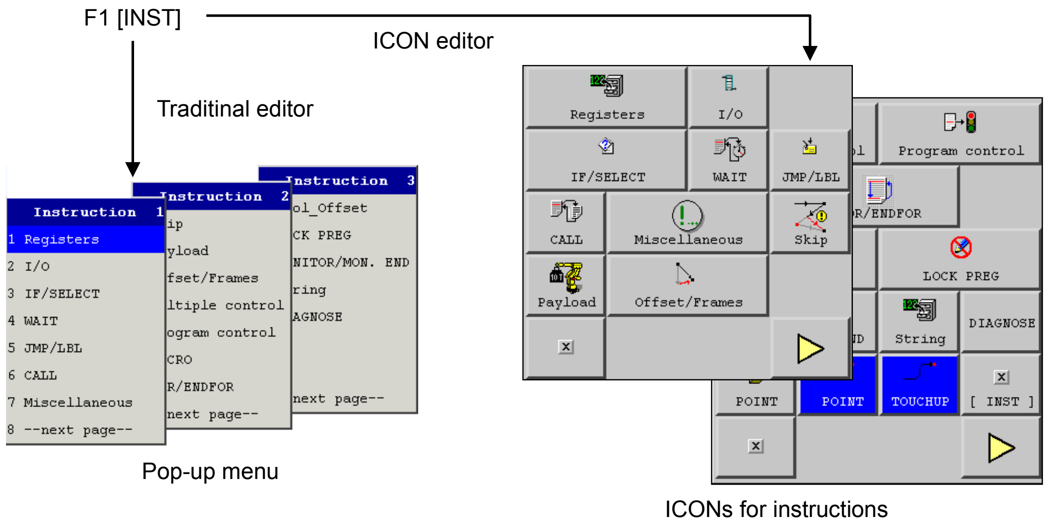


Fig. 11.3.1(b) ICONs for instructions

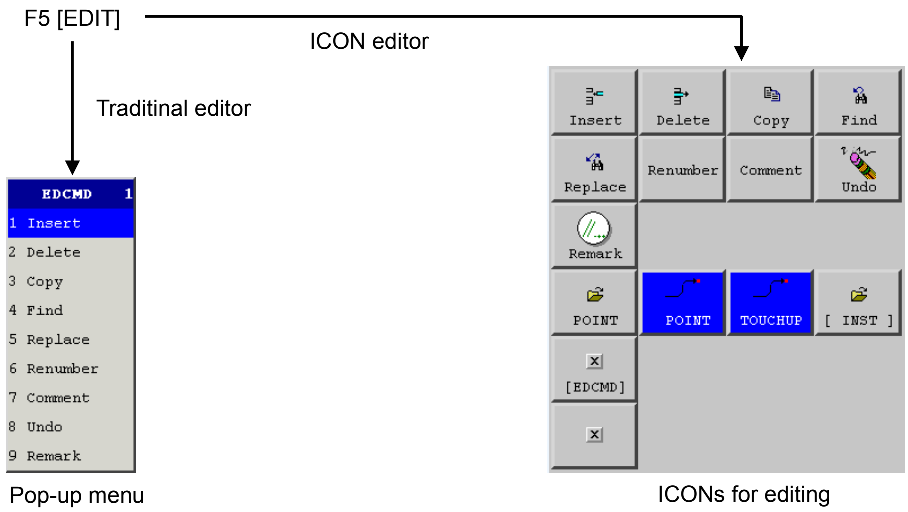


Fig. 11.3.1(c) ICONs for editing

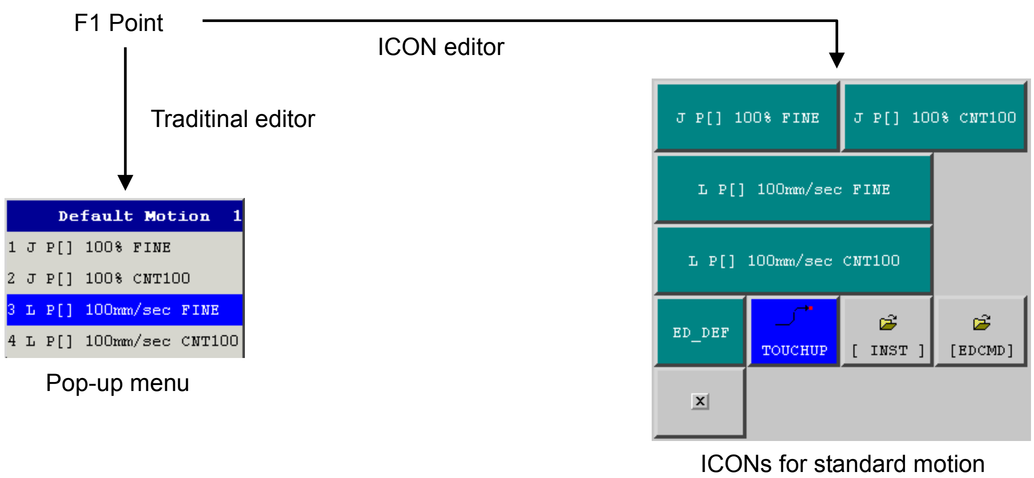


Fig. 11.3.1(d) ICONs for standard motion

11.4 CUSTOMIZATION FUNCTIONS

11.4.1 Overview

It is possible to customize various screen on the teach pendant. This Subsection describes the method to customize the screens on the teach pendant.

11.4.2 Background Color

It is possible to change the background color of a screen by the procedure 11-1. In initial setting, the background color is set to light blue.

> ⚠ **CAUTION**
> There isn't *i*Pendant Color Setup screen in R-30*i*B Plus controller.

Procedure 11-1 Setting the background color

Step

1. Press [MENU] key
2. Select SETUP.
3. Press F1, [TYPE].
4. Select *i*Pendant Setup. The *i*Pendant General Setup screen is displayed.
5. Select *i*Pendant Color Setup. The following screen is displayed.

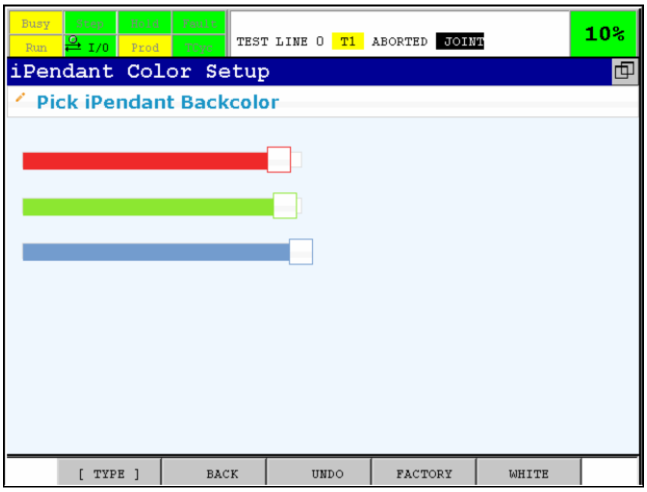


Fig. 11.4.2 *i*Pendant color setup

6. Select a bar by up and down arrow key, then specify the color rate by left and right arrow key. In case that the teach pendant equips the touch panel, the color rate can be specified by touching the points on each bar.
7. In order to redo the setting, press F3, UNDO key.
8. In order to return the background color to light blue of the initial setting, press F4, FACTORY key.
9. In order to set the background color to white, press F5, WHITE key.

> ⚠ **CAUTION**
> It takes longer time to display *i*Pendant Color Setup screen than other screens.

> ⚠ **CAUTION**
> In order to display the text on the screen clearly, the color which verges on black cannot be set as the background.

11.4.3 Setting up the HMI Screen

When press MENU key on teach pendant, instead of a default screen menu, it is possible to display the screen that is made with HTM or STM files, as a customize screen menu (HMI menu). It is possible for the user to customize the HMI menu.

The HMI menu can set different screens for full menu, quick menu and remote menu.

When a HMI full menu is set up, if [MENU] key is pressed, the following screen will be displayed.

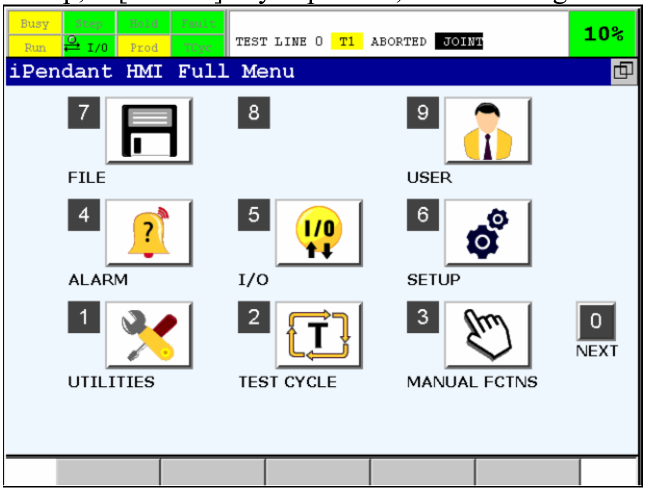


Fig. 11.4.3 (a) *i*Pendant HMI full menu

When a HMI quick menu is set up, if [FCTN] key is pressed and "QUICK/FULL MENUS" is selected in the displayed menu, the following screen will be displayed.

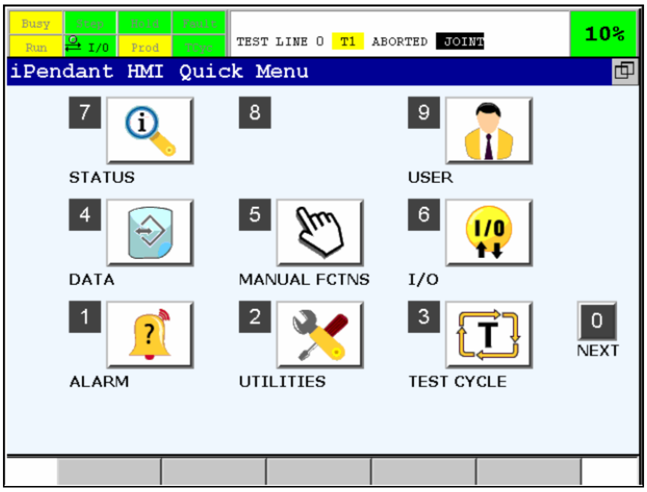


Fig. 11.4.3 (b) *i*Pendant HMI quick menu

To create a user-defined HMI full menu or quick menu, use Procedure 11-2. To define and set up a HMI full menu, quick menu and remote menu use Procedure 11-3.

Procedure 11-2 Creating a user-defined HMI full menu or quick menu

Step

1 Press [MENU] key
2 Select SETUP.
3 Press F1, [TYPE].
4 Select *i*Pendant Setup. The following screen will be displayed.

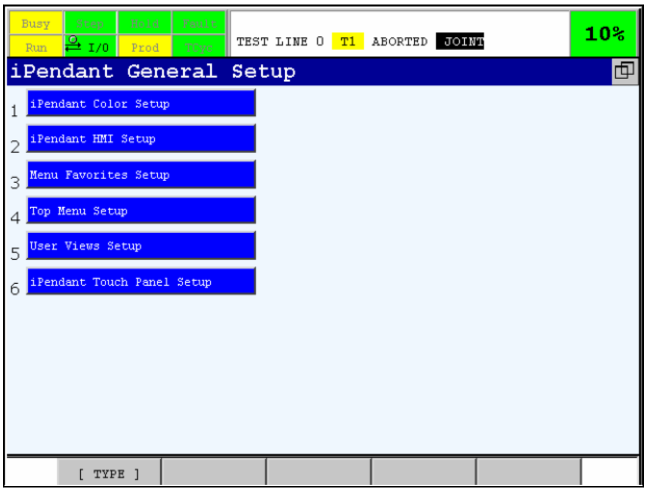


Fig. 11.4.3 (c) *i*Pendant setup menu

5 Select "*i*Pendant HMI Setup". The following screen will be displayed.

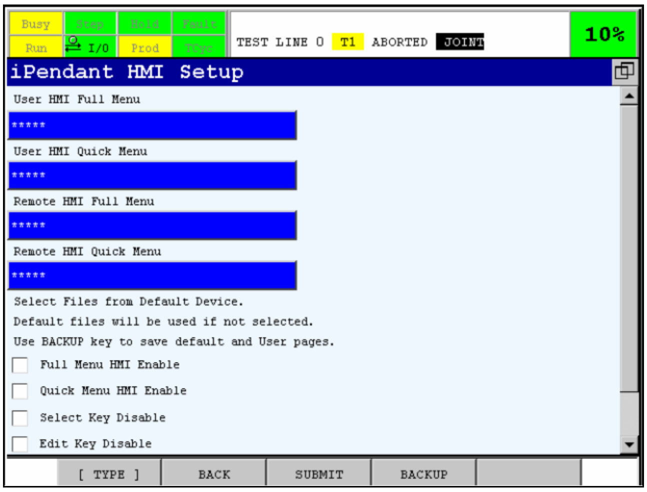


Fig. 11.4.3 (d) *i*Pendant HMI setup menu

6 If F4, BACKUP is pressed, the STM files are saved to the currently selected file I/O unit. (refer to 8.1.) The content of the standard HMI full menu is saved in JIPHMIFL.STM, and the content of the standard HMI quick menu is saved in JIPHMIQK.STM. For detail about the files saved by F4, BACKUP, refer to the procedure 11-3.
7 Move the saved files in a personal computer. Edit this file and can create a user original HMI full menu and HMI quick menu. Add or delete the menu item in files, as required.

> ⚠ **CAUTION**
> The HMI full menu file (IPHMIFL.STM) must have "*i*Pendant Setup" as a menu item. If passwords can be used on the controller, the HMI quick menu file (IPHMIQK.STM) must have "Passwords" as a menu item. Keep these menu items if required. If the menu items are missing, created menu files do not operate properly.

8 Restore the files to the controller.
9 To set up and use the created files, use Procedure 11-3.

Procedure 11-3 Setting up selections on an HMI full menu, quick menu, and remote menu

Step

1. Press [MENU] key.
2. Select SETUP.
3. Press F1, [TYPE].
4. Select *i*Pendant Setup. Fig. 11.4.3 (c) will be displayed.
5. Select "*i*Pendant HMI Setup". Fig. 11.4.3 (d) is displayed.
6. Select the HTM or STM file used by a full menu, quick menu, remote full menu, and remote quick menu.
   a. Click the blue bar below the menu to be set up. The list of the HTM and STM files on the selected file I/O unit is display.
   b. Select the desired file and press the [ENTER] key.
   c. Continue to select files for individual menus ([User HMI Full], [User HMI Quick], [Remote HMI Full], and [Remote HMI Quick]).
   d. Press F3, SUBMIT. The name displayed on the blue bar is erased to indicate that setup is completed.
7. To enable a selected full or quick HMI menu mode or remote full or quick HMI menu mode, click the check box next to the selection (Full Menu HMI Enable or Quick Menu HMI Enable) located at the lower part of the screen and press F3, SUBMIT.
8. To disable the [SELECT], [EDIT], or [DATA] key in HMI mode, or to enable F1, [TYPE] key in HMI mode, scroll down and click the check box next to the desired key name. For example, the following screen is displayed.

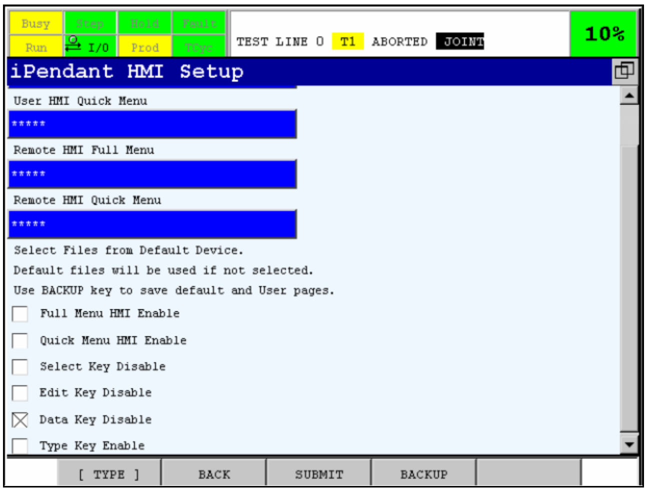


Fig. 11.4.3 (e) *i*Pendant HMI setup screen, if the [DATA] key is to be disabled

After clicking the check box, press F3, SUBMIT. After F3, SUBMIT is pressed, if the “Submit completed” is displayed in status window, enable the current status.

> ⚠ **CAUTION**
> Each time the HMI setup screen appears, a check mark appears and the current settings are indicated.

9. To save all HMI files in the currently selected file I/O unit, press F4, BACKUP. The following files are backed up.

IPHMIQK.STM	The standard contents of the HMI quick menu using the text link
IPHMIFL.STM	The standard contents of the HMI full menu using the text link
JIPHMIQK.STM	The standard contents of the HMI quick menu using icon
JIPHMIFL.STM	The standard contents of the HMI full menu using icon

The following submitted files are back up.

USRHMIQK.STM	The registered contents for user HMI quick menu
USRHMIFL.STM	The registered contents for user HMI full menu
REHMIQK.STM	The registered contents for remote HMI quick menu
REHMIFL.STM	The registered contents for remote HMI full menu

10 To display previous menu, press F2, BACK.

11.4.4 Menu Favorites

By using Menu Favorites, it is possible to generate and save a list of the most frequently displayed menus.

To set a Menu Favorites list, use Procedure 11-4. To change a Menu Favorites list, use Procedure 11-5.

The list of the registered menu is automatically reflected to ICON menu for favorite screen. When press MENU key, ICON menu for favorite screen is displayed at the bottom of a screen. (For details, refer to “ICON menu” in the Subsection “2.3.1 Teach pendant”.)

Procedure 11-4 Adding a menu to the Menu Favorites list

Step

1 Display the menu to be added as Menu Favorites menu.
2 Press and hold down SHIFT and press the DISP key.
3 Select Menu Favorites.
4 Select "Add Current". The currently displayed menu is added to the list.
See Fig. 11.4.4(a).

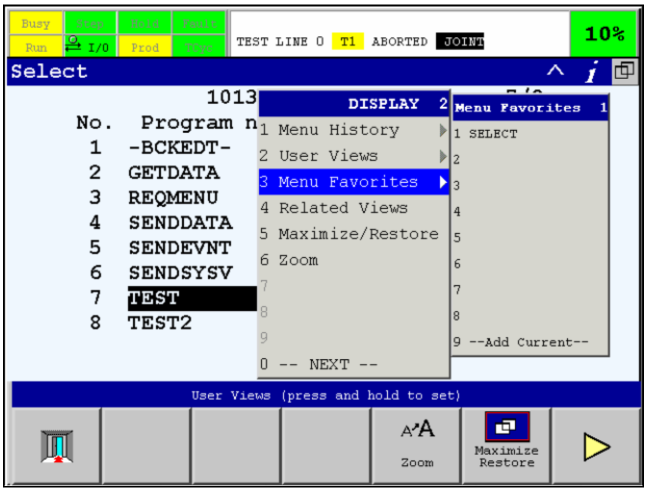


Fig. 11.4.4 (a) Adding to menu favorites

5 To display a menu saved in the Menu Favorites list, press and hold down Shift and press the DISP key. Move the cursor to Menu Favorites and select a menu from the list.

Procedure 11-5 Changing Menu Favorites

Step

1 Press [MENU] key
2 Select SETUP.
3 Select F1, [TYPE].
4 Select *i*Pendant Setup.
5 Select Menu Favorite Setup. The following screen will be displayed.

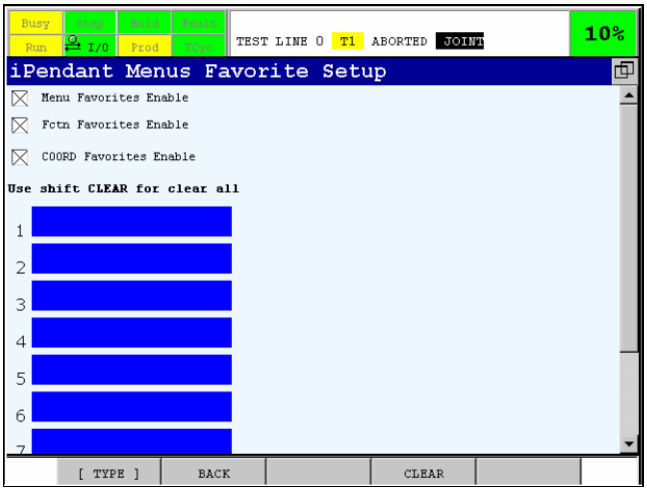


Fig. 11.4.4 (b) Changing menus favorites setup screen

6 To enable/disable ICON menu for favorite screen, check or uncheck the check box of "Menu Favorites Enable". If "Menu Favorites Enable" is unchecked, ICON menu is not displayed.
7 To enable/disable ICON menu for sub-function, check or uncheck the check box of "Fctn Favorites Enable". If "Fctn Favorites Enable" is unchecked, ICON menu is not displayed. But depending on the application tool software, there is not ICON menu for sub-function. (Refer to "ICON menu" in the Subsection "2.3.1 Teach pendant".)
8 To enable/disable ICON menu for coordinate system, check or uncheck the check box of "COORD Favorites Enable". If "COORD Favorites Enable" is unchecked, ICON menu is not displayed.
9 To erase one item from the Menu Favorites list, move the cursor to the item to be erased and press F4, CLEAR. The Menu Favorites list is automatically redisplayed.
10 To erase all items from the Menu Favorites list, press Shift and F4, CLEAR. The Menu Favorites list is automatically erased.
11 To display the main *i*Pendant General Setup screen, press F2, BACK.

11.4.5 Top Menu Setup

It is possible to setup the top menu which is displayed by pressing MENU key while the *i* key is pressed.

Up to 10 top menus can be defined. Each top menu can be switched by the function keys. Up to 9 icons can be allocated in a top menu.

Select the function key and enter the label name in each page according to the procedure 11-6. Set up the window configuration and the screen displayed by the ICON, and the image of the ICON for the ICONs allocated in a page according to the procedure 11-7.

Procedure 11-6 Setting a page in the top menu

Step

1 Press [MENU] key
2 Select SETUP.
3 Select F1, [TYPE].
4 Select *i*Pendant Setup. The *i*Pendant General Setup screen is displayed.

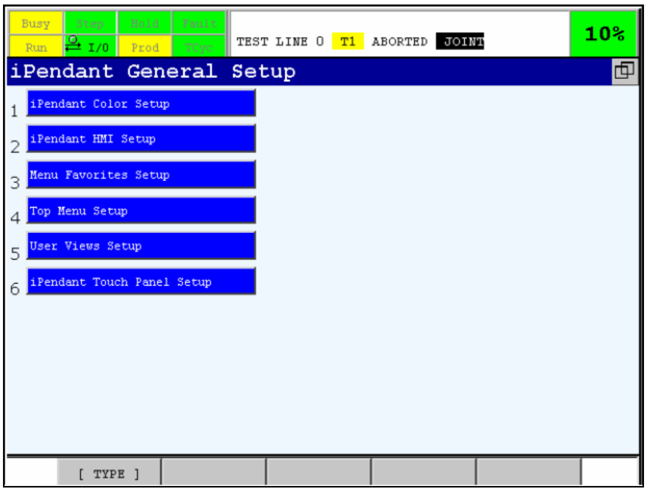


Fig. 11.4.5 (a) *i*Pendant general setup screen

5 Select Top Menu Setup. Top Menu Setup screen is displayed.

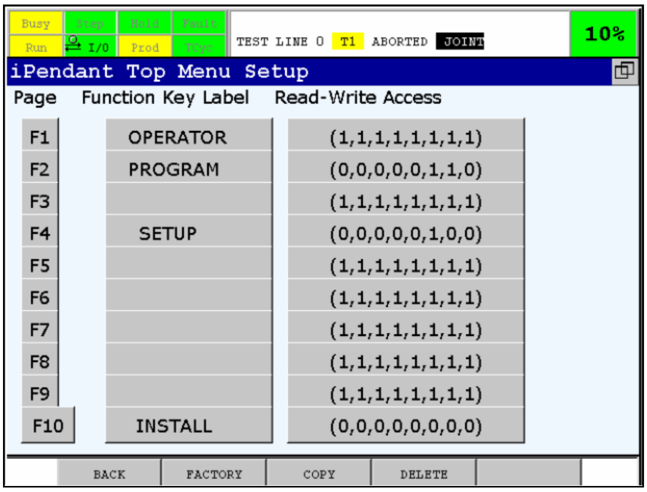


Fig. 11.4.5 (b) Top menu setup screen

6 In order to set the label name, select the Function Key Label button, and then enter the label name.
In order to copy the label name, press F3, COPY, then enter the source and destination page to copy.
In order to delete the label name, press F4, DELETE, then enter the function key number to delete.
In order to initialize all setting, press F2, FACTORY.

Procedure 11-7 Setting an ICON in top menu

Step

1 Display the Top Menu Setup screen by same procedure as the step 1 to 5 in procedure 11-6.
2 Select the page that ICONs are setup by up and down arrow key, then press the [ENTER] key.

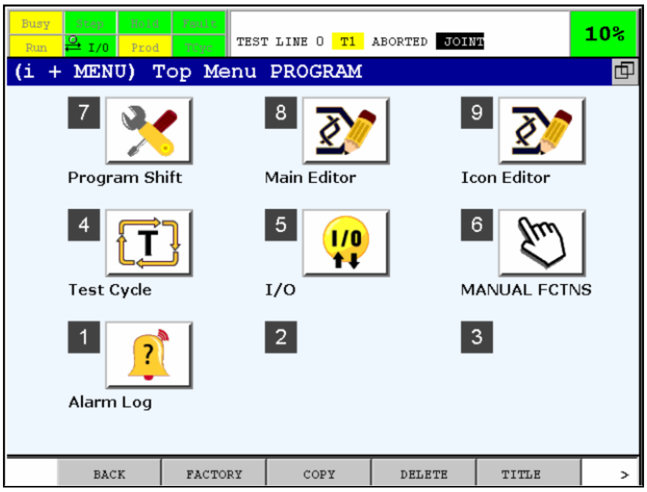


Fig. 11.4.5 (c) ICON setup screen 1

3 In order to initialize all setting, press F2, FACTORY.
In order to copy the ICON, press F3, COPY, then enter the source and destination number to copy.
In order to delete the ICON, press F4, DELETE, then enter the ICON number to delete.
In order to change the title of page, press F4, TITLE, then enter the title of the page. The title of the page is displayed in the blue bar at the upper of the screen.

If the [NEXT] key is pressed, the next screen is displayed.

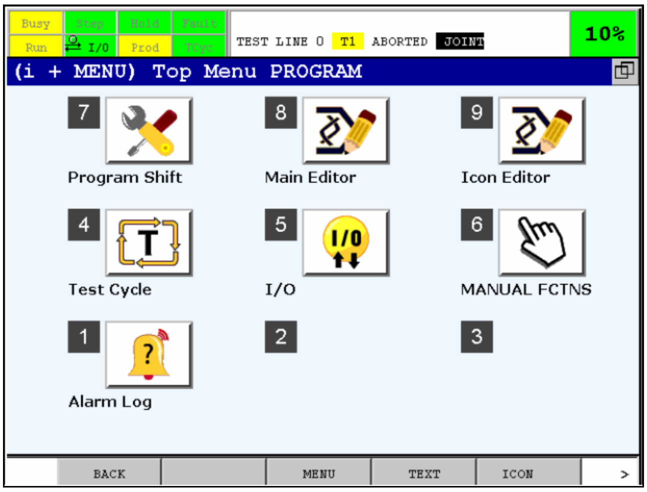


Fig. 11.4.5 (d) ICON setup screen 2

4 In order to add ICON or modify setting of ICON, press F3, MENU. If the ICON number is entered, the following screen is displayed.

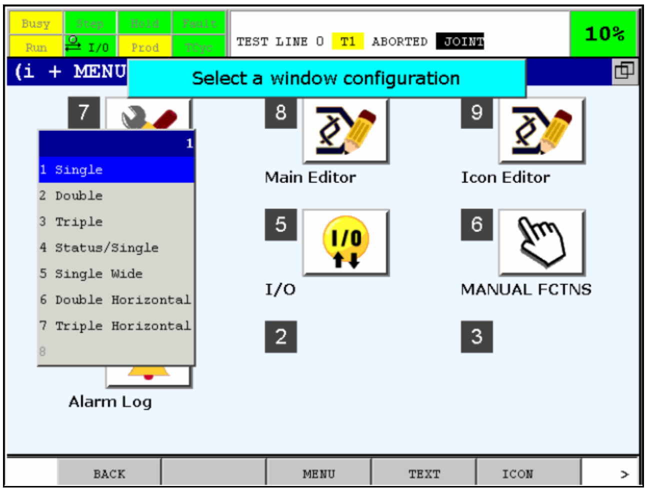


Fig. 11.4.5 (e) Display setup

Select a window configuration, then press the ENTER key. The screen list is displayed.

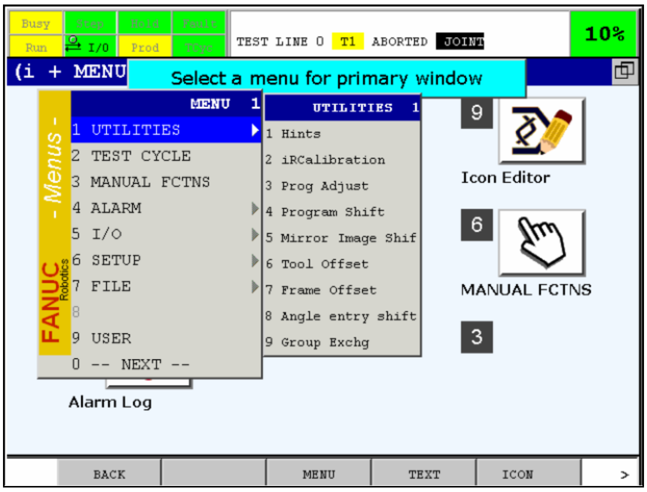


Fig. 11.4.5 (f) Screen setup

Select a screen from the list, then press the [ENTER] key. In case that multiple screen is selected in the display setup, it is need to select the same number of screens as the number of division of the screen.

5 In order to modify the name of ICON, press F4, TEXT and enter the number of the ICON to modify, then enter the name of ICON.

6 In order to modify the image of ICON, press F5, ICON and enter the number of the ICON to modify. The images of ICON created by the user in FR: are displayed. Move the cursor to the image of ICON that you want to select, then press F4, SELECT. In order to cancel the operation, press F5, CANCEL.

11.4.6 User Views

By using the screen User Views function, it is possible to save up to eight User Views statuses of frequently used user-defined single window or multi-window displays. For example, if an alarm screen, file screen, and I/O screen are usually displayed using a 3-screen window, this set of screens can be defined as screen User Views. When saved, the screen User Views is assigned a name, which is displayed as a User Views item on the display menu, so that User Views can be redisplayed by selecting it later. The User Views described earlier is displayed as Alarm | FILE | Cell I in the User Views list.

To add a set of menus as User Views, use Procedure 11-8. To change a User Views list, use Procedure 11-9.

Procedure 11-8 Adding User Views

Step

1 On the teach pendant, display the set of screens to be added as User Views.
2 Press and hold down SHIFT and press the DISP key.
3 Select "User Views".
For example, the screen below will be displayed.

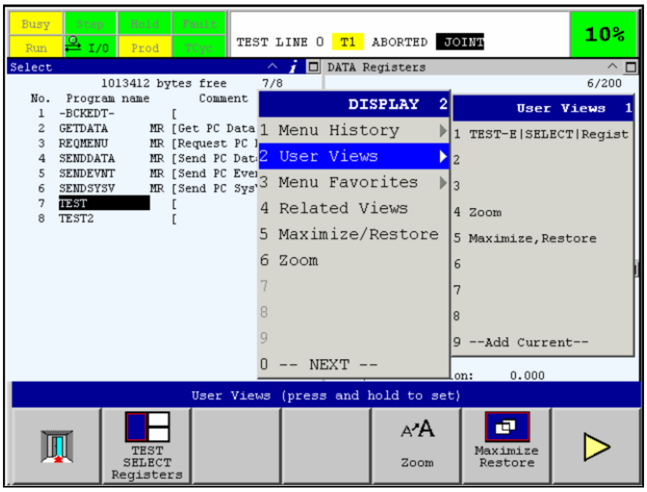


Fig. 11.4.6 (a) Adding user views

4 Select "Add Current". The currently displayed screen set is added to the list as screen name (for a 1-screen display), screen name|screen name (for a 2-screen display), or screen name|screen name|screen name (for a 3-screen display).
5 For each User Views to be added, repeat steps 1 to 4.
6 To display User Views saved in the User Views list, press and hold down Shift and press the DISP key. Move the cursor to User Views and select the User Views to be displayed from the User Views list.

Procedure 11-9 Changing User Views

Step

1 Press [MENU] key
2 Select SETUP.
3 Press F1, [TYPE].
4 Select *i*Pendant Setup. The *i*Pendant General Setup screen will be displayed.

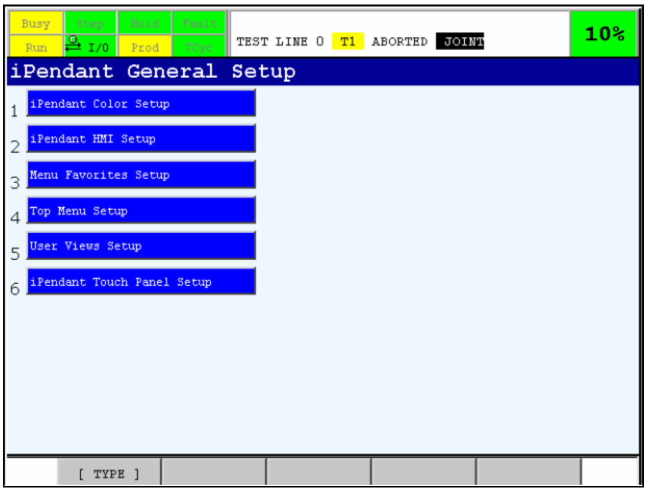


Fig. 11.4.6 (b) *i*Pendant general setup screen

5 Select User Views Setup. For example, the following screen will be displayed.

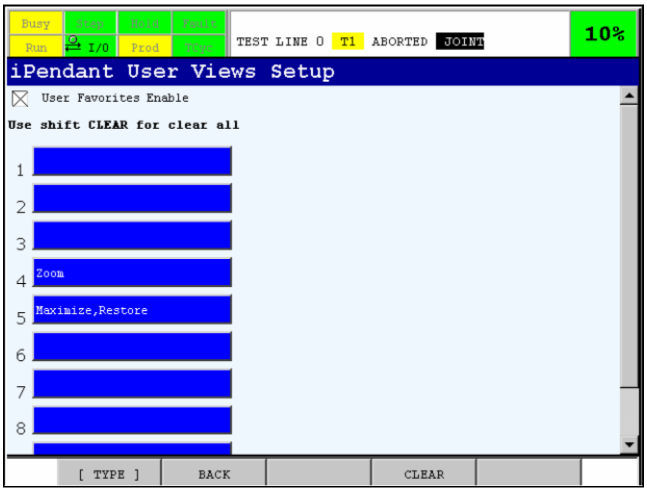


Fig. 11.4.6 (c) *i*Pendant user views setup screen

6 To enable/disable ICON menu for User View, check or uncheck the check box of "User Favorites Enable". If the check box of "User Favorites Enable" is unchecked, ICON menu is not displayed.
7 To erase one item from the User Views list, move the cursor to the User Views to be erased and press F4, CLEAR. The User Views list is automatically redisplayed.
8 To erase all items from the User Views list, press Shift and F4, CLEAR. The User Views list is automatically erased.
9 To display the main *i*Pendant General Setup screen, press F2, BACK.

11.4.7 Setting up *i*Pendant Touch Panel

This is screen to restrict part of operation by touch panel of the teach pendant.

Procedure 11-10 Setting up the touch panel

Step

1 Press [MENU] key
2 Select SETUP.
3 Press F1, [TYPE].
4 Select *i*Pendant Setup. The *i*Pendant General Setup screen will be displayed.

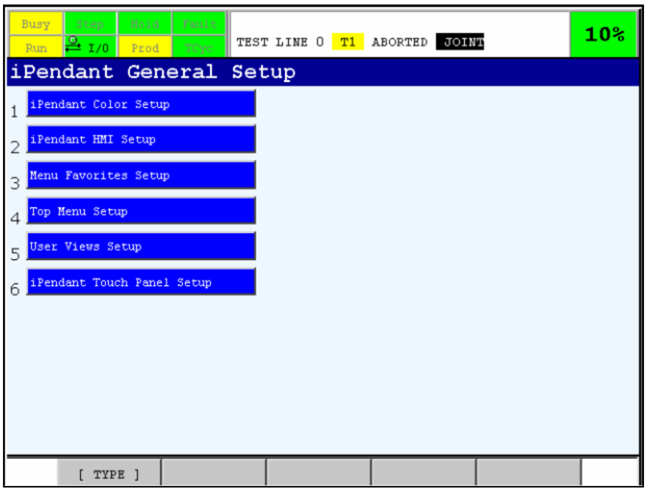


Fig. 11.4.7 (a) *i*Pendant general setup screen

5 Select *i*Pendant Touch Panel Setup. A screen similar to following figure is displayed.

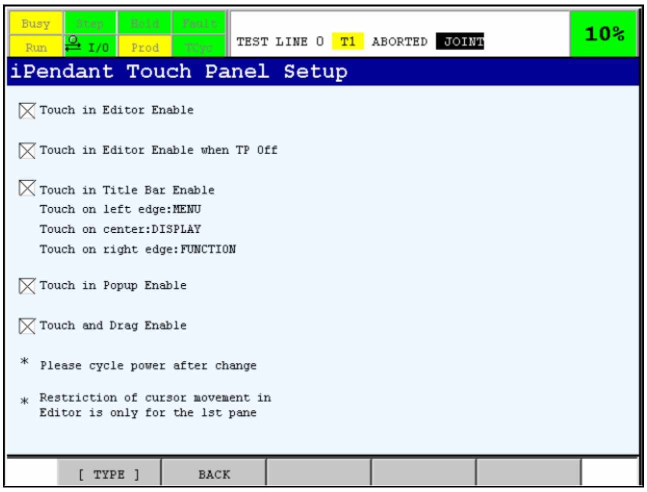


Fig. 11.4.7 (b) *i*Pendant touch panel setup screen

If check box is checked, corresponding item is enabled. If unchecked, the item is disabled.

Table 11.4.7 Setup items in *i*Pendant touch panel setup screen

Item	Description
Touch in Editor Enable	When this function is enabled, if the screen is touched in the program edit screen, the cursor moves to the touched point. This item does not affect scroll by drag.
Touch in Editor Enable when TP Off	This item specifies whether the function of “Touch in Editor Enable” is enabled when TP is Off. If “Touch in Editor Enable” is disabled, this setup item is neglected. In this case, you cannot move cursor when TP is disabled even if this item is checked.
Touch in Title Bar Enable	This is function to display menu when you touch title bar (blue bar which displays title). The left side corresponds to MENU key, and the center corresponds to SHIFT + DISP key, and the right side corresponds to [FCTN] key. This item also effects on Maximize/Restore icon on the title bar.
Touch in Popup Enable	When this function is enabled, you can select items in a menu by touch.
Touch and Drag Enable	When this function is enabled, the following two operations become possible by the drag of finger on the touch panel. - scroll text and Web page - zoom in the 4D graphics display screen, and parallel shift and rotation of the view of point.

NOTE

“Touch in Editor Enable” and “Touch in Editor Enable when TP Off” have effect on only the first pane. If screens are in double pane mode, the items effects only on the left pane. You can move cursor by touch in the right pane regardless of setting of the items.

NOTE

“Touch in Editor Enable” and “Touch in Editor Enable when TP Off” do not affect the scroll by the function of “Touch and Drag Enable”. In order to restrict the scroll by the operation on the touch panel, it needs to uncheck the check box of “Touch and Drag Enable”.

NOTE
Cycle power is needed for change to take effect. When you change status of check box, “Please cycle power after change” is displayed on status window of the teach pendant.

11.4.8 Setting up *i*Pendant Brightness

It is possible to change the brightness of a screen by the procedure 11-11.

⚠ **CAUTION**
There isn’t *i*Pendant Brightness Setup screen in R-30*i*B and R-30*i*B Mate controller.

⚠ **WARNING**
Change the brightness of *i*Pendant screen to HIGH, the temperature of *i*Pendant rises. When you use in high temperature environment, please be cautious.

Procedure 11-11 *i*Pendant Brightness Setup

Step

1 Press [MENU] key
2 Select SETUP.
3 Press F1, [TYPE].
4 Select *i*Pendant Setup. The *i*Pendant General Setup screen will be displayed.

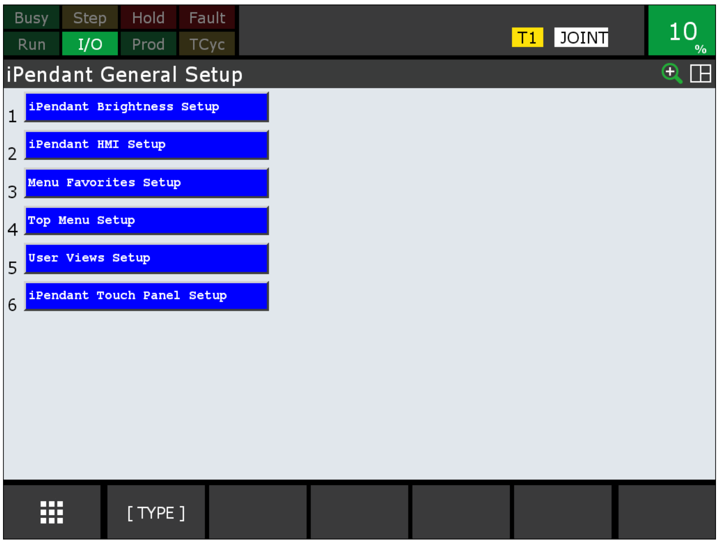


Fig. 11.4.8 (a) *i*Pendant general setup screen

5 Select *i*Pendant Brightness Setup. A screen similar to following figure is displayed.

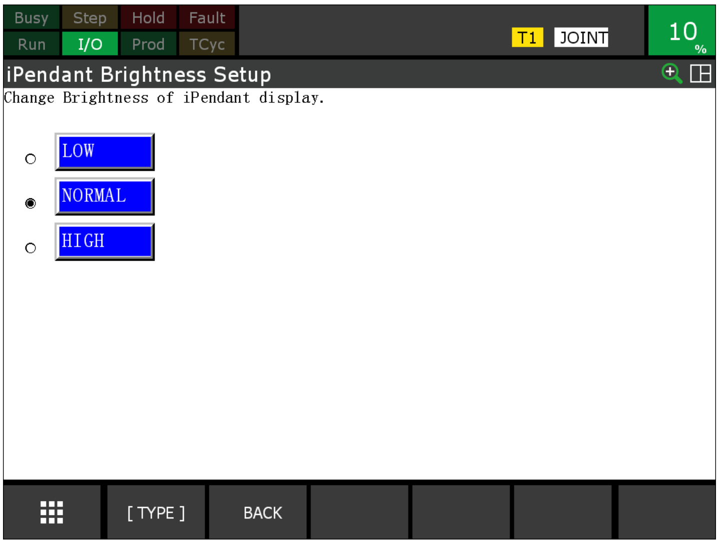


Fig. 11.4.8 (b) *i*Pendant brightness setup screen

6 To change the brightness of *i*Pendant, select from LOW, NORMAL and HIGH. The brightness is automatically changed after selection.

7 To display the main *i*Pendant General Setup screen, press F2, BACK.

11.4.9 History

The history list records the eight mostly recently displayed menus. This list is automatically generated and cannot be changed. When the DISP key is pressed while the SHIFT is pressed and "0 –NEXT– " is selected, the history list is displayed. When a name in the list is selected, the screen with that name will be displayed.

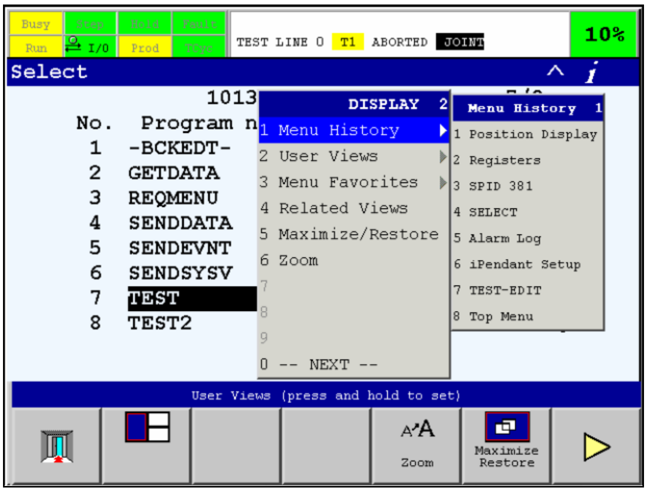


Fig. 11.4.9 History screen

11.5 OPERATION PANEL

11.5.1 Overview

Operation panel function is only available if R651 ”FRL Params” (Standard setting software option) is ordered. This function is not supported with R650 ”FRA Params” (North America setting).

Operation panel function is the software panel for various operation on the teach pendant.

Operation panel has following things.
- Lamps indicating the input and output signal status
- Buttons setting output signal by manual
- Register display window
- Buttons setting register value

Followings can be set.
- Signal type and signal number assigned to a lamp and a button
- Color of lamps indicating ON condition
- Label of button
- Register number, whose value is displayed to register display window
- Register value assigned to a button

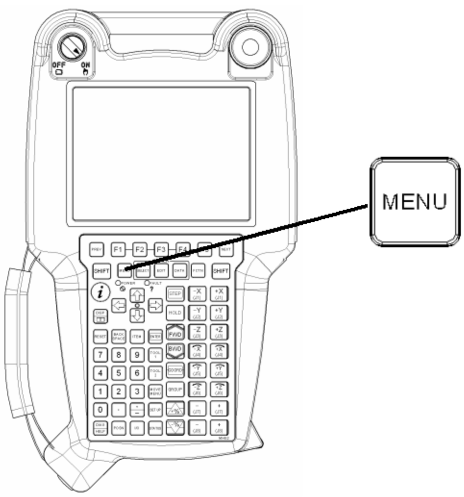
One operation panel can display one of the following 3 types of screen.
- 16 I/O showing lamps, 16 I/O setting buttons
- 14 I/O showing lamps, 14 I/O setting buttons
- 1 Register display window, 10 register setting buttons, 4 I/O showing lamps, 4 I/O setting buttons

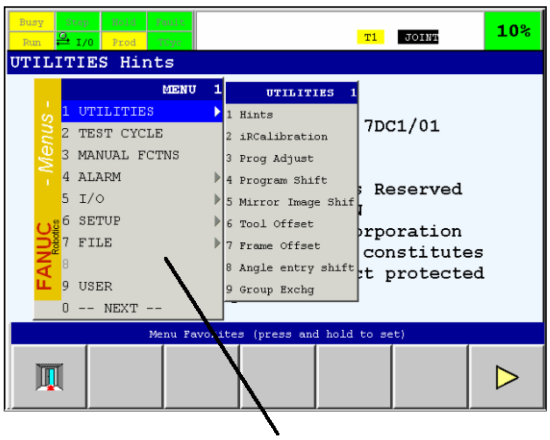
Operation panel can be made up to 4 screens.

> ⚠**WARNING**
> Do not use operation panel function for a switch impacting human health or equipment damage. Safety switch must be used for a switch impacting human health.

11.5.2 Basic Operation

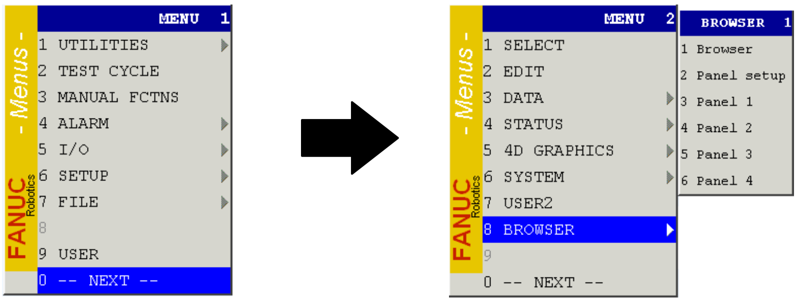
To display the operation panel screen, press MENU key to display the screen selection menu.





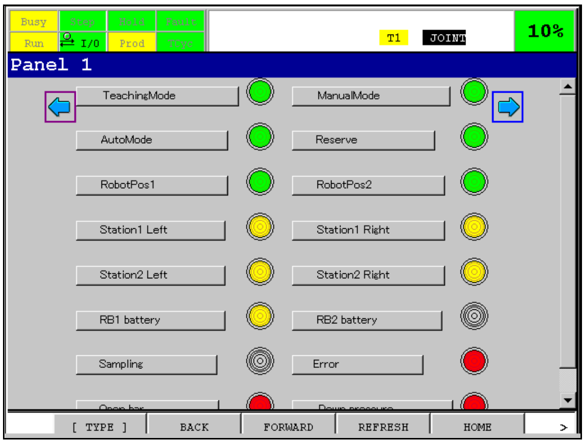
Screen selection menu

Select 0: NEXT item in the screen selection menu to display next page of the menu, and select BROWSER item. Then the BROWSER menu will appear at the right side of the menu.



Up to four operation panels can be created and the name of each panels are "Panel 1", "Panel 2", "Panel 3" and "Panel 4". Before creating a panel, only the item "Panel setup" exists in the pop-up menu. After creating a panel, the created panel will be added into the pop-up menu. You select a panel name in the pop-up menu by pressing right arrow key and move cursor to submenu, then you can display the selected panel on a screen.

Operation panel has several lamps indicating the input signal status and several buttons output the signals by manual. Additionally panel change buttons are also located.

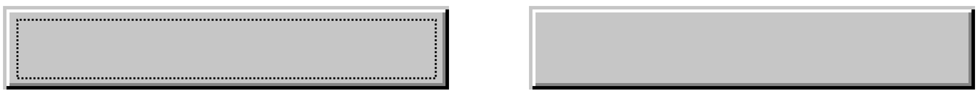


- Lamp
  Turn ON or OFF to indicate the input signal status. Input signal of DI, DO, RI, RO, WI, WO, UI, UO, SI or SO can be asserted to a lamp.

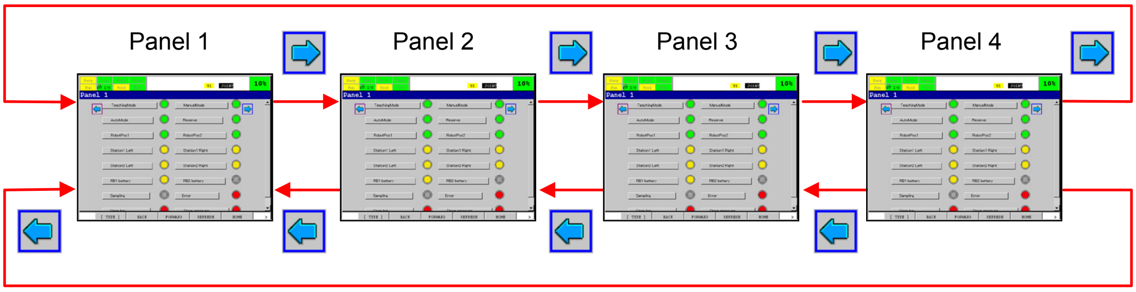
Red lamp Amber lamp Green lamp Lamp: Turn OFF

- Button
  By pressing a button, an output signal can be turned ON or OFF alternately by manual, or a register value can be modified. On the teach pendant, select a button by using arrow keys and press the [ENTER] key, or touch a button in case that the touch panel is enabled, then the output signal assigned to the selected button will be turned ON or OFF, or the specified register value will be set. In the selected button, the frame drawn by dot-line appears as follows.

Selected button No selected button



- Panel change button
  At the both side of panel top area, panel change buttons are prepared. Pressing this button, another panel will appear on a screen.



Each panel can be selected directly in the F1, [TYPE] menu.

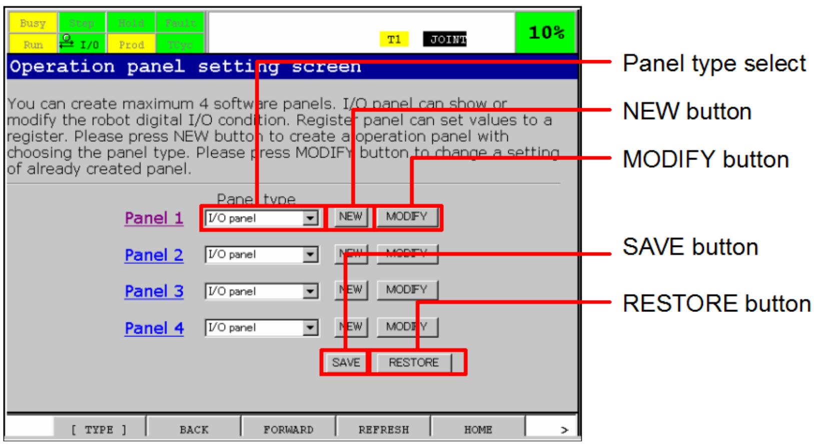
```
# TYPE          1
1 Browser
2 Panel setup
3 Panel 1
4 Panel 2
5 Panel 3
6 Panel 4
```

11.5.3 Panel Setup

Before using operation panel, it is necessary to set panel according to system.

11.5.3.1 Operation panel setting screen

To start setting operation panel screen, select "Panel setup" item from the F1, [TYPE] menu of Internet screen. Then operation panel setting screen as below is displayed.



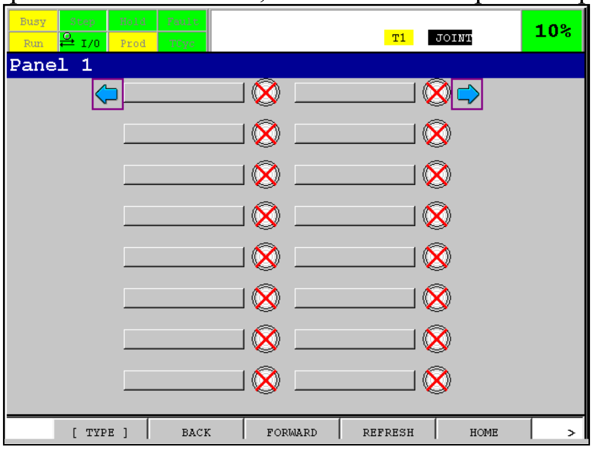
- Panel type select
  I/O panel, I/O panel (7 lines) and register panel is available.
- NEW button
  When you want to create a new panel, at first select a panel type and press this button. When you have created a panel but re-create again as a new panel then select a panel type and press this button also.
- MODIFY button
  When you want to change the setting of a created panel, please press this button.
- SAVE button

After panel setting, all setting data can be saved to an external memory by pressing this button. In series V8.10, saved to memory card. In other series, saved to the selected device, which is selected by MENU button -> "FILE" -> F5 key "UTIL" -> "Set Device".

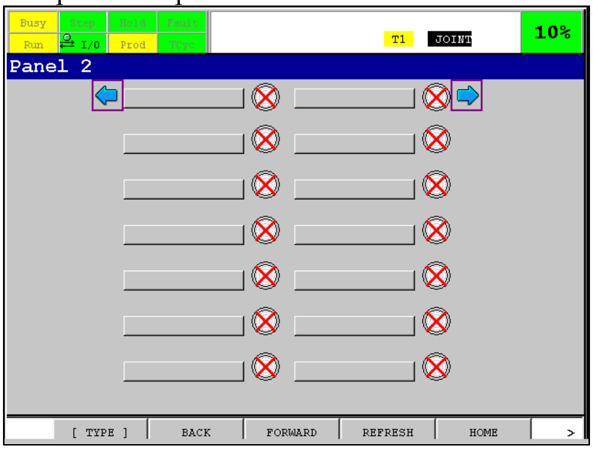
- RESTORE button
  All saved setting data can be restored from a selected device (memory card in V8.10) by pressing this button.

As an operation panel, the following three kind of panel are supported.

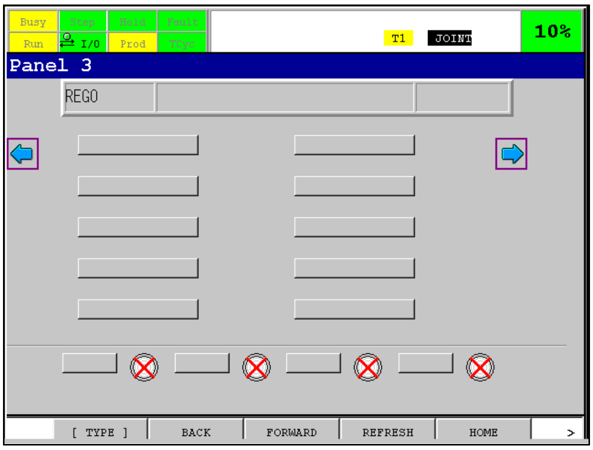
- I/O Panel
  This panel has 16 buttons for signal output and 16 lamps for signal input. This panel is useful for the status monitor panel of production execution, and the manual operation panel for external devices.



- I/O panel (7 lines)
  One line is eliminated to expand the space of each line.



- Register panel
  Enter the specified data into a register by pressing a button of this panel. This is useful to select the kind of production workpiece. Entered value will be used for branching the execution in robot programs.



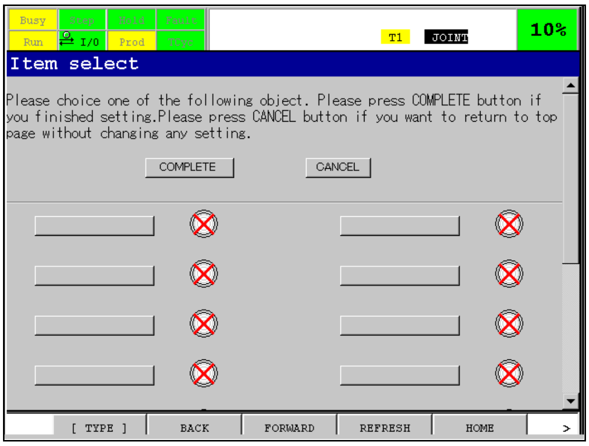
To select a panel type, move a cursor on the field of a panel type on the panel setting screen, and press the [ENTER] key, or touch the field of a panel type in case that the touch panel is enabled. Then the following selection menu will be displayed. You can select an item in this menu by using arrow keys, or by touching an item in case that the touch panel is enabled.

```
I/O panel
I/O panel(7 lines)
Register panel
```

11.5.3.2 Item select screen

When you enter a NEW button or a MODIFY button on the panel setting screen, the following item select screen is displayed. From this screen, you need to select a button or a lamp you need to set, then press the [ENTER] key, or touch a button or a lamp you need set in case that the touch panel is enabled.

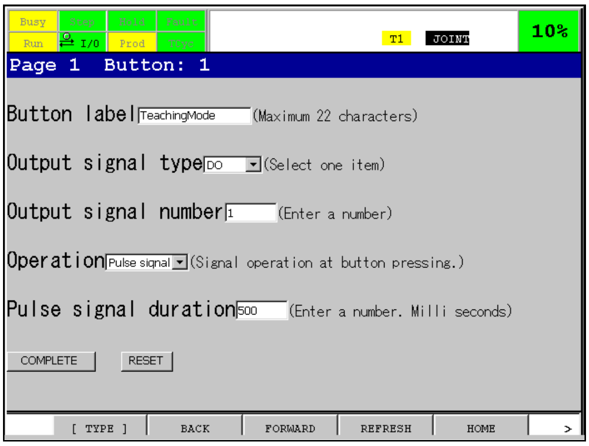
If you want to return to the previous screen without any changing, press CANCEL button. Then any changing is ignored and goes to the top page of operation panel.



11.5.3.3 Item setting

This Subsection explains setting of output signal change button and signal display lamp. Refer to Section 11.5.3.7 about register display window and register value setting button.

When a button is selected, the following button setting screen will be displayed. Move a cursor to an item you need to setup and press the [ENTER] key, or touch an item you need to set up in case that the touch panel is enabled.



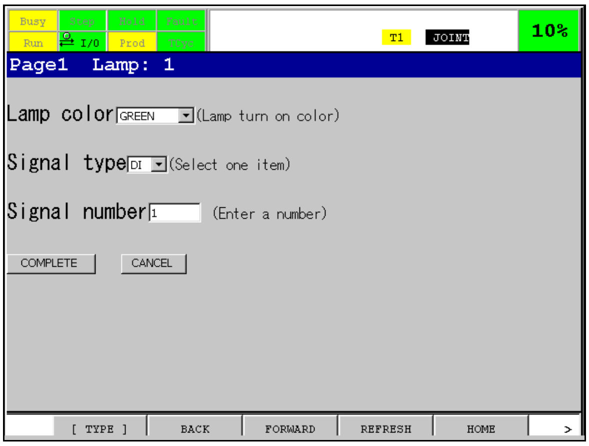
- Button label
  This is the label string displayed on a button. Up to 22 characters are used as a label.
- Output signal type
  DO signal or RO signal can be selected. If "No use" is selected as the signal type, only label string is displayed as the string and the button is not displayed.
- Output signal number
  Specify the output signal number. If 0 is specified, no signal will be output at pressing the button.
- Operation
  Specify the output signal transition at pressing a button. If ALTER is specified, the output signal will be reversed from the previous status at pressing the button.
- Pulse signal duration
  If Pulse signal is specified in the operation column, you can specify the duration of pulse signal output. Unit is milliseconds. When you press a pulse signal button, output signal will be asserted and keep ON for the specified duration, and will be turned OFF automatically.

After setting completion, press COMPLETE button. If you want to cancel all changing done on the screen, press RESET button then all setting values on the setting screen will be returned to previous values.

NOTE about pulse signal

- Precise time pulse signal can not be realized by this function. The duration of signal output can be overdue.
- Maximum setting time of pulse signal duration is 9000 milliseconds. If a enter value is over from the maximum time, the duration time will be cut to 9000 milliseconds.
- During signaling pulse signal, other signal output operation can not work.

When a lamp is selected, the following lamp setting screen will be displayed.



- Lamp color
  RED, AMBER, and GREEN can be selected as the color indicating the signal ON status. If RESERVED is selected, the lamp with cross bar marker will be always displayed. If NO USE is selected, the lamp will not be displayed.
- Signal type
  Specify the type of an input signal.
- Signal number
  Specify the number of an input signal.

After setting completion, press COMPLETE button. If you want to cancel all changing done on the screen, press RESET button then all setting values on the setting screen will be returned to previous values.

NOTE

- The item which has the downward triangle button on the right hand of the input window has a select list. When the cursor is moved to the input window and press the [ENTER] key, or touch the input window on the touch panel, a select list is displayed. Select an item from the displayed list by using **left and right (not up and down)** arrow keys and press the [ENTER] key, or touch an item in the displayed list on the touch panel.

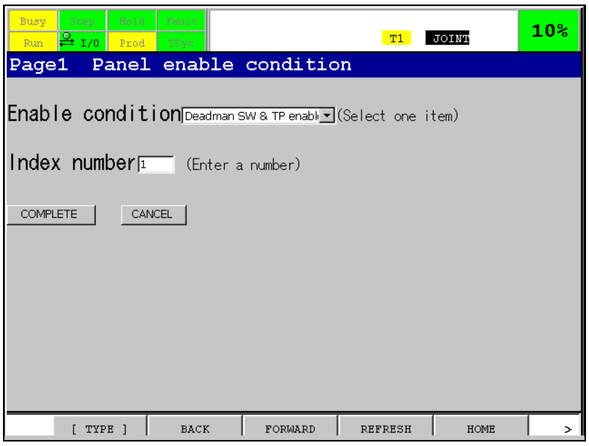
NOTE

- When entering into the text box using the software keyboard, make sure that the cursor is displayed in the text box, and then start up the software keyboard.

11.5.3.4 Panel enable condition setting

You can use a specific signal or TP ENABLE or deadman switch to enable the operation from an operation panel. The following "Setup panel enable condition" button exists at the bottom of the Item select screen by scrolling to down side.

Press this button then you will get the following screen to setup the panel enable condition.



The followings are supported conditions for enabling the operation from an operation panel.

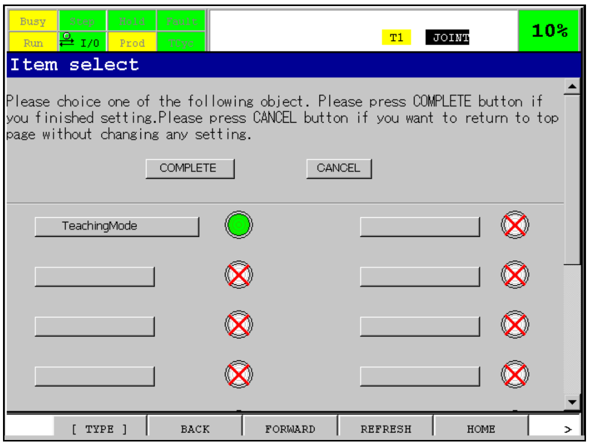
- DI, DO, RI, RO
  When a specified input signal is ON, the operation from a panel is enabled. If you need to use this condition, enter a signal number into the index number item. If you enter 0 into the item, this condition is disabled.
- TP enable
  When a TP enable switch is ENABLE, the panel operation is enabled.
- Deadman Switch
  When a deadman switch is grasped, the panel operation is enabled.
- Deadman SW & TP enable
  When a TP enable switch is ENABLE and a deadman switch is grasped, the panel operation is enabled.
- No condition
  Panel enable condition is not specified and always the panel operation is enabled.

NOTE About panel enable condition

- Lamps allocated on a panel are independent with the panel enable condition setting and always functional.
- Only one condition is usable for each panel.
- When press a button in disable condition, the button pressing action will occur but in this case, signal will not output. For confirmation of signal output, we recommend preparing a lamp to confirm signal output and at button operation, operator should check the lamp.

11.5.3.5 Finish setting

On item setting screens, press COMPLETE button, then the item select screen will be displayed. On this screen, you can confirm the setting of lamps and buttons. If you need to continue the setting, you will select another lamp or button.

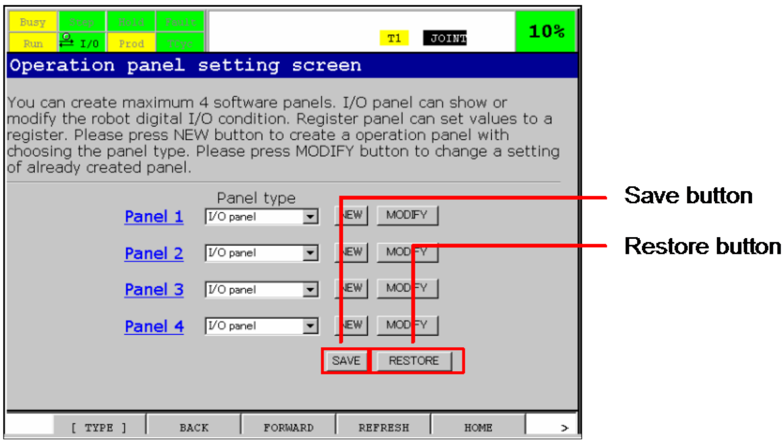


When all setting is finished, press COMPLETE button on the item select screen. All setting values are stored into your controller non-volatile memory and return to the top page of operation panel function.

If you want to cancel all setting, press CANCEL button on the item select screen. All setting values are abandoned and return to the top page of operation panel function.

11.5.3.6 Save and restore setting value

Setting data of all operation panels can be stored to an external memory by pressing a SAVE button on the top page of operation panel function. In series V8.10, saved to memory card. In other series, saved to the selected device, which is selected by MENU button -> "FILE" -> F5 key "UTIL" -> "Set Device". We recommend saving the setting data continually.



The following files are stored to an external memory when the SAVE button is pressed.

- panel1.stm HTML file for a panel 1
- panel2.stm HTML file for a panel 2
- panel3.stm HTML file for a panel 3
- panel4.stm HTML file for a panel 4
- panel1.dt Setting data for a panel 1
- panel2.dt Setting data for a panel 2

- panel3.dt Setting data for a panel 3
- panel4.dt Setting data for a panel 4

The above files can be stored by the all file store on FILE screen.

These saved files can be restored when the RESTORE button is pressed.

Note about the restore

In cases below, pay attention when setting contents is restored.

- After setting contents are restored, when the restored operation panel is not listed in the menu displayed by F1, [TYPE], the restored operation panel is listed by the procedure 11-12.
- When the operation panel which is made in R-30*i*A controller or R-J3*i*B controller is restored, it is necessary to perform the procedure 11-12 to convert the operation panel to the format of R-30*i*B controller.

Procedure 11-12 Special operation after restore

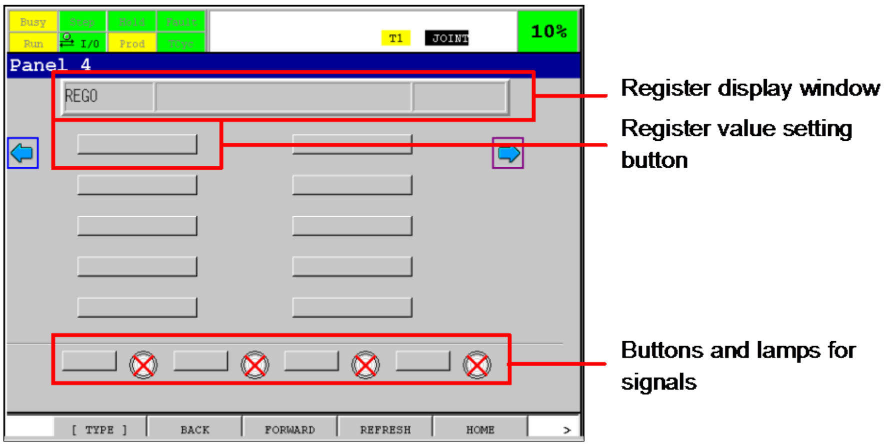
Step

Perform the operation below on all operation panels from panel 1 sequentially.

1. Press “MODIFY” button in the operation panel setting screen to display the item select screen.
2. Press “COMPLETE” button in the item select screen.

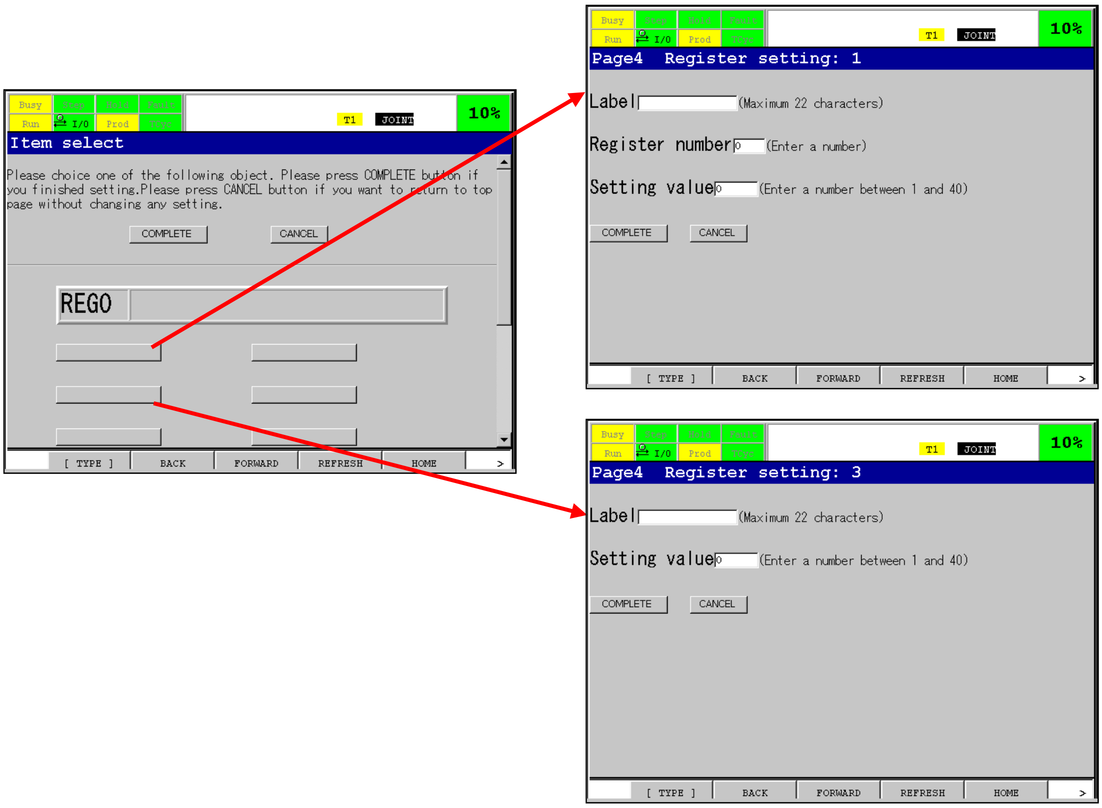
11.5.3.7 Register panel

Register panel can set a specified value to a register when a button is pressed. One register panel can set a value to just only one register. Register panel also has a window for displaying the current value of the register. Register panel also has some buttons and lamps for signal monitoring and setting. But the button is smaller than the I/O panel and maximum character of a button label is 10.

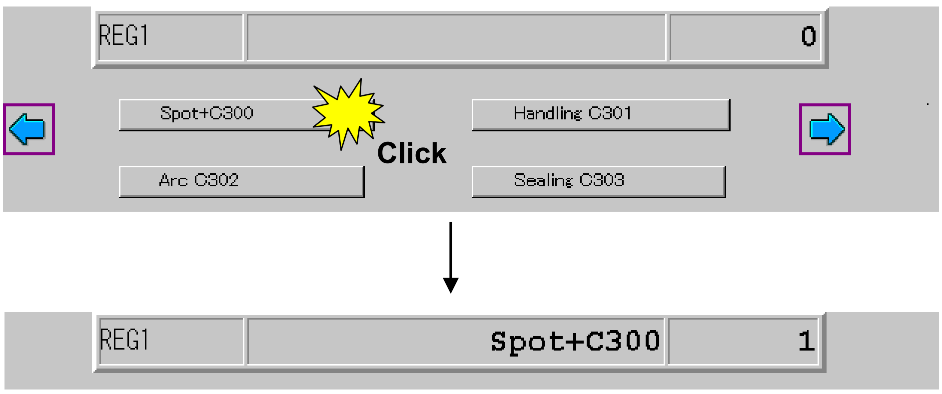


Register panel can be created by the same manner as an I/O Panel. Select register panel as the type of a panel on the top page of the setting screen.

The label string and the setting value, from 1 to 40, should be specified for each buttons. The register number can be specified in the setting screen only for the upper left button.

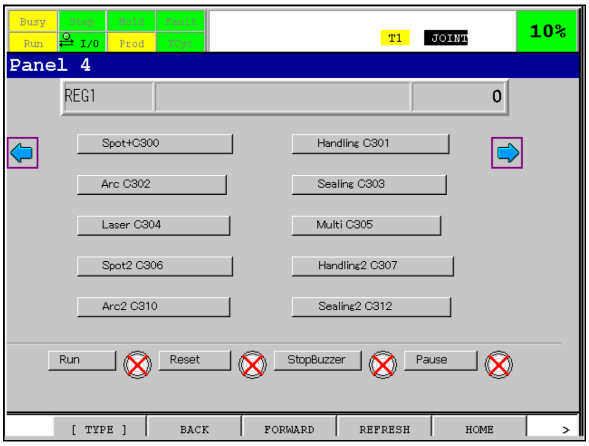


When a button allocated on a register panel is pressed, the specified setting value will be set to the specified register. And the current value of the register is displayed in the register display window.



When a register value is changed by not only the operation from the button of a register panel, but also other method, for example the program instruction execution or the operation from the register data screen, the label of a button, that setting value matches the current register value, will be displayed in the register display window.

Register panel can be used to select the kind of production goods. The setting value is used for branching the program execution in a PNS/RSR program.



The followings are sample program to select a sub program based on the value of a register.

```
SELECT   R[1]=1, CALL C300
         =2, CALL C301
         =3, CALL C302
         =4, CALL C303
         =5, CALL C304
         =6, CALL C305
         =7, CALL C306
         =8, CALL C310
         =9, CALL C312
         ELSE, JMP LBL[100]
END
LBL[100]
UALM[1]
```

If R[1] is 1, then call subprogram C300.
Followings are similar.

R[1] value does not match with either value, go to LBL[100]

Assert a user defined alarm.

11.5.4 Restriction

- By the quick button operation, operation will become invalid. Please perform the button operation slowly and surely.
- When you allocated an output signal for a button, please set the lamp for confirmation on a panel. And, at the time of the button operation, please confirm the state of the lamp by all means.
- The operation panel function supports Japanese and English. The displayed language of the operation panel setting screen is changed according to the selected language. When the selected language is changed, please cycle power of the controller. In case that the selected language is changed to the language other than Japanese, the operation panel made in Japanese screen is not listed correctly in the menu displayed by pressing F1, [TYPE] key. In that case, do the Procedure 11-13 to display the menu correctly.

Procedure 11-13 Special operation after change of the language

Step

Do the following operation to display the menu by F1, [TYPE] correctly.

1. Press “MODIFY” button on the operation panel setting screen to display item select screen.
2. Press “COMPLETE” button on item select screen.

12 *i*HMI

This chapter describes the functions and operations of *i*HMI.
*i*HMI is a user interface with a design that has been unified with CNC manufactured by FANUC. The minimum required functions have been collected together on the home screen, so that even beginners can start up robots easily. You can perform basic robot setup and create simple programs just by performing operations as given in the guides. You can also access useful functions such as graphical monitor screen display and backup from the home screen.

Contents of this chapter

12.1 OVERVIEW OF THE HOME SCREEN AND BASIC OPERATIONS ON THE GUIDE SCREENS

This section gives an overview of the home screen and describes the basic operations on the guide screens.

12.1.1 Overview of the Home Screen

The home screen displays the flow from the initial setup of the robot to program creation and execution, separating it into the groups "Setup," "Teach," "Run," and "Utility." There is a guide screen that allows you to carry out selection and input for setting operations such as "Basic Setup" and "Create a Program" in an easy-to-understand manner while referring to guide information displayed for each step. If you tap an icon on the home screen, the corresponding guide screen will be displayed.

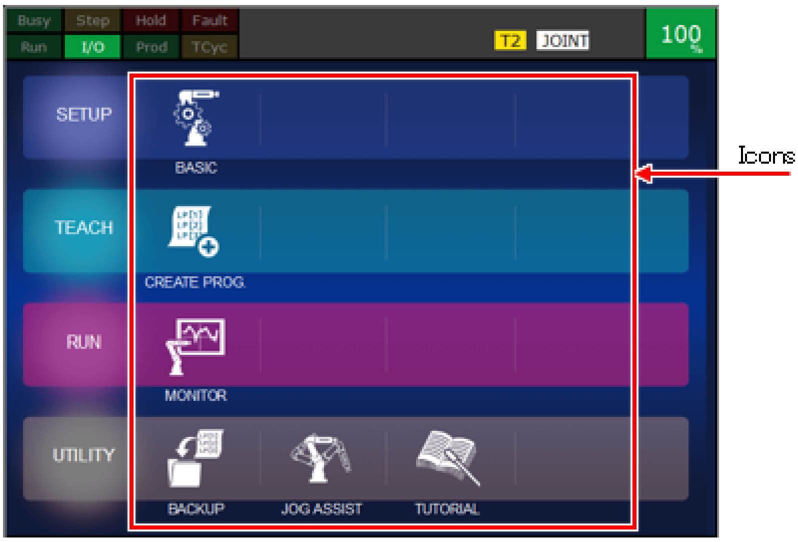
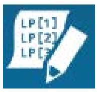


Fig. 12.1.1 Home screen

The table below shows information about the guide screen and functions that are displayed by tapping the icon.

Table 12.1.1 Icons on the home screen

Type	Icon	Description
Setup	 Basic	Displays the Basic Setup Selection screen. On the selection screen, you select the Initial Setup screen and End of Arm Tool Setup screen. On the Initial Setup screen, you set up times and networks that are needed to start using the robot. On the End of Arm Tool Setup screen, you set up tool information so that the robot can move tools.
Teach	 Create Program	Displays the Create a Program screen. On the Create a Program screen, you create programs using template programs (sample programs) for achieving certain objectives. You can also select a "Blank Template" for creating a program from scratch.
	 Select Program	Displays the Select a Program screen. The "Select Program" icon is displayed if multiple programs have already been created. If you tap this icon, the Select a Program screen will be displayed.
	 Name of currently selected program	Displays the Edit a Program screen. An icon with the name of the program that is currently selected is displayed. If you tap this icon, the Edit a Program screen will be displayed.
Run	 Monitor	Displays the Production Monitor screen. On the Production Monitor screen, the production running status is monitored. The display can also be changed to 4D graphic or to the current position screen.
Utility	 Backup	Displays the File Backup screen. On the File Backup screen, you select devices to which backups will be made, select files and data to backup, and execute backups.
	 Jog Assist	Displays the Jog Assist screen. On the Jog Assist screen, you set the override and coordinate system settings that are related to jog operation through icon selection operations.
	 Tutorial	Displays the Tutorial Selection screen. On the Tutorial screen, you select the Jog a Robot screen, Program Edit screen and Program Execution screen. The tutorials enable you get experience of and learn about basic operations such as robot jog operation and program creation.

12.1.2 Basic Operations on the Guide Screens

This section describes the guide screens that are displayed through icon operations on the home screen.
The following content is explained:

- Components of the guide screens
- Display on the guide screens
- Examples of step operations on the guide screens

12.1.2.1 Components of the guide screens

The figure below shows an example of a guide screen display (example: Initial Setup screen) and the name of each part.

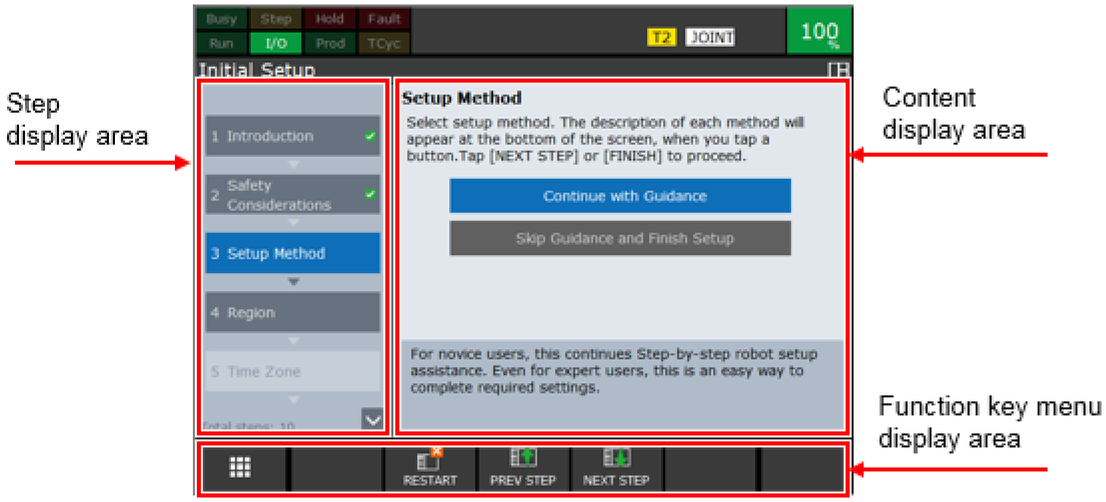


Fig. 12.1.2.1 (a) Guide screen example

Step display area

Displays the operation steps for confirmation and settings to be performed on the guide screen. The number of steps displayed varies depending on the type of the guide screen. Subsequent steps may also be added or changed, depending on the setting content.

For steps for which setup or confirmation has been completed, if you tap [Next Step] in the function key menu display area, the current step will be marked with a check and you will move on to the next step. The selected step is displayed in blue.

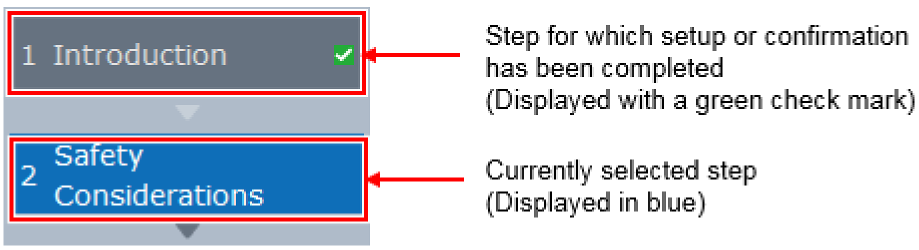


Fig. 12.1.2.1 (b) Guide screen example

If you tap or in the step display area, the step display will scroll.

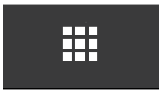
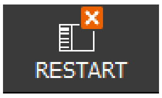
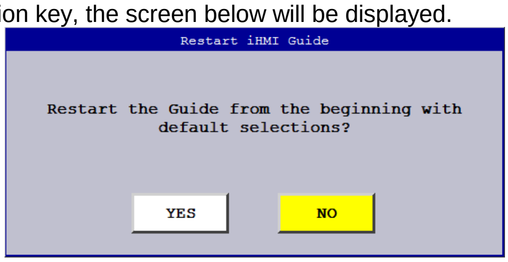
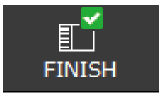
Content display area

Displays content information (descriptions and setting items) corresponding to the selected step.

Function key menu display area

Displays the function keys that are used on the guide screen.
The table below shows the types and functions of the function keys.

Table 12.1.2.1 Guide screen function keys

Function key	Description
	Displays the home screen.
	The current guide will restart. If you tap this function key, the screen below will be displayed.  If you want to restart, tap "Yes." All changes will be canceled and the screen will return to the original screen. If you do not want to restart, tap "No." The guide will not restart.
	Goes back to the previous step. Tap this function key when you want to go back to a step that is before the current step.
	Proceeds to the next step. After confirmation and settings for the current step have been finished, tap this function key and go on to the next step. This function key is not displayed when there are items in the content display area that need to be set but have not yet been set. Perform the necessary settings in the content display area.
	Finishes the settings on the guide screen. This function key is displayed in the last step on the guide screen. If you tap this function key, the setting content for each step is finalized and the screen returns to the original screen.
	Displays a description screen. If this function key is displayed, tap it to display a more detailed description of the selected step.

12.1.2.2 Basic operations on the guide screens

This section describes the basic operations on the guide screens.

NOTE
What happens if you tap the [Basic] icon on the home screen is described as an example of guide screen operation. As the Production Monitor screen and the Jog Assist screen are not in step guide format, please refer to "12.4 Operations related to Run" and "12.5.2 Jog Assist" for the operations on them.

Procedure 12-1 Example of guide screen operation

Step

1 Press the [MENU] key and select "Utility" → "Home" from the screen menu that is displayed. The home screen will be displayed.

Fig. 12.1.2.2 (a) Home screen

NOTE
1 If initial setup is not finished, the Initial Setup screen is displayed by selecting the [MENU] key → "Utility" → "Home." Perform initial setup by following the steps displayed on the Initial Setup screen.
2 The home screen can also be displayed by tapping in the function key area.

2 Tap the [Basic] icon on the home screen.

Fig. 12.1.2.2 (b) "Basic" icon

The Basic Setup Selection screen will be displayed.

NOTE
In cases where tapping the icon directly displays the target function screen (cases where there are no levels), the selection screen will not be displayed.

3 Tap [Initial Setup].

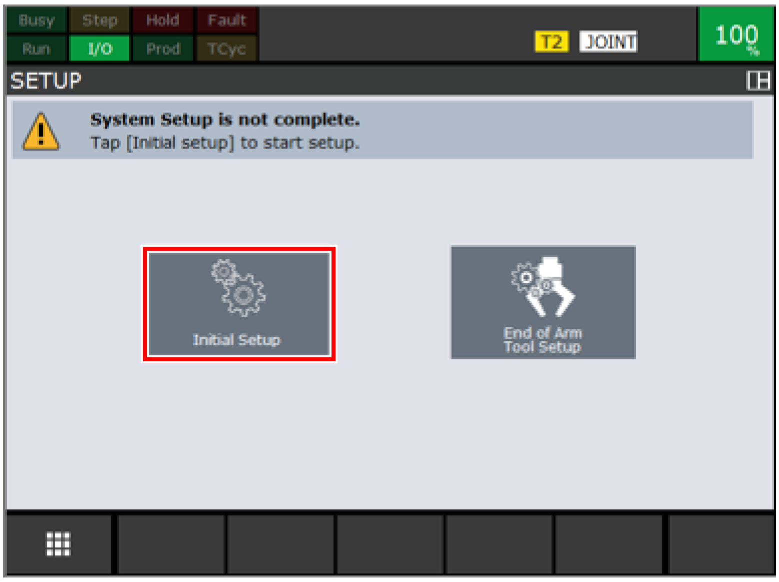


Fig. 12.1.2.2 (c) Basic Setup Selection screen

The Initial Setup screen will be displayed.

4 Confirm or set the content displayed in the content display area, and tap [Next Step].

Fig. 12.1.2.2 (d) Initial Setup screen

The step in the step display area moves on to the next step, and the content in the content display area will change.

Fig. 12.1.2.2 (e) Initial Setup screen (step 2)

5 Once again, confirm or set the content for the step, and tap [Next Step].
The step in the step display area moves on to the next step, and the content in the content display area will change.

NOTE
Repeat this operation until the last step.

6 Confirm the setting (selection) content in the last step and tap [Finish].

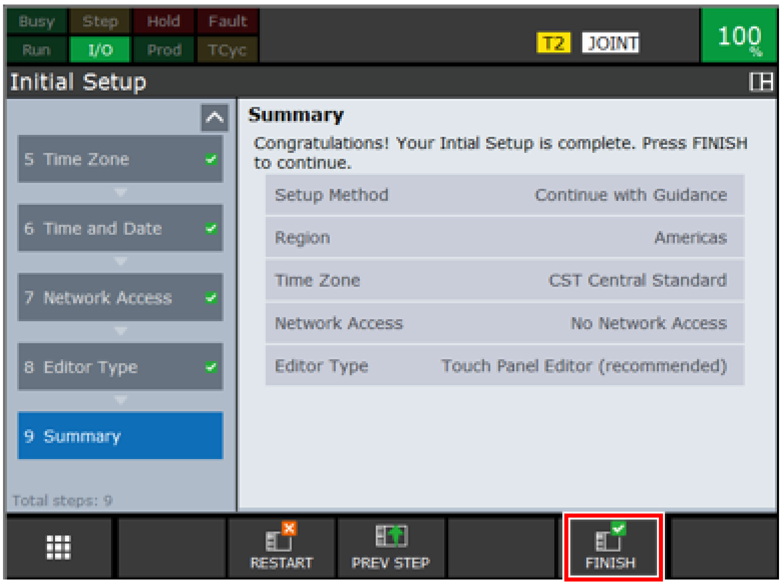


Fig. 12.1.2.2 (f) Initial Setup screen (last step)

The content that has been set is finalized. Initial setup finishes, and the screen returns to the original screen (Basic Setup Selection screen).

12.2 OPERATIONS RELATED TO SETUP

Tap the [Basic] icon on the home screen and display the Basic Setup Selection screen.
On the Basic Setup Selection screen, select either the Initial Setup screen or the End of Arm Tool Setup screen.
The screen display flow for setup is shown below.

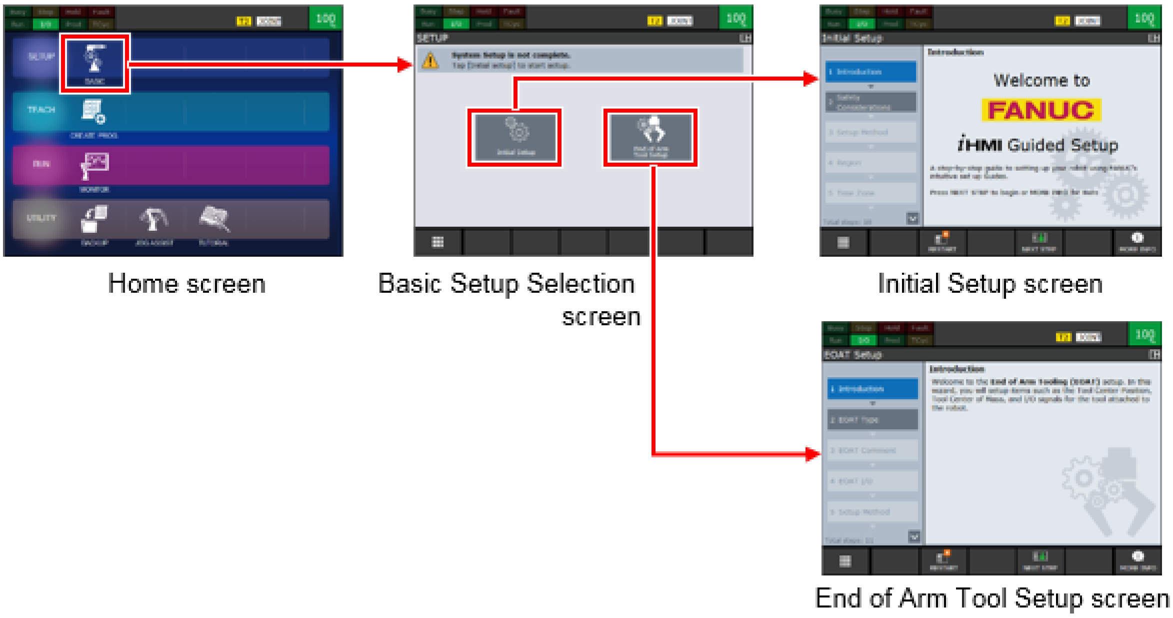


Fig. 12.2 Screen display flow (Setup)

12.2.1 Initial Setup

On the Initial Setup screen, you perform initial setup such as times and networks that are needed to start using the robot.
The Initial Setup screen is shown below. For the basic operations on the guide screens, refer to "12.1.2 Basic operations on the guide screens."

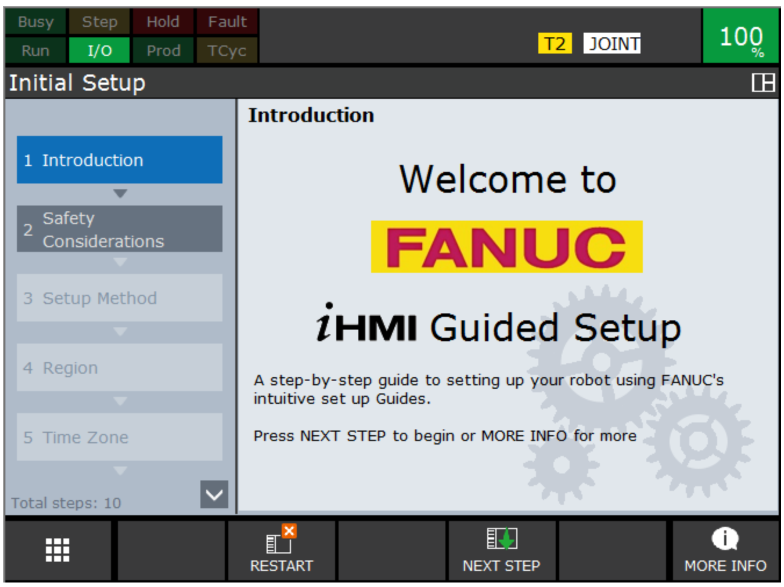


Fig. 12.2.1 Initial Setup screen

The main setup items on the Initial Setup screen are as follows:
- Setup Method
- Region
- Time Zone
- Time and Date
- Network Access
- Editor Type

12.2.2 End of Arm Tool Setup

On the End of Arm Tool Setup (EOAT) screen, you set items such as the tool center position (TCP), tool center of mass, and I/O signals for the tool attached to the robot.
The EOAT Setup screen is shown below. For the basic operations on the guide screens, refer to "12.1.2 Basic operations on the guide screens."

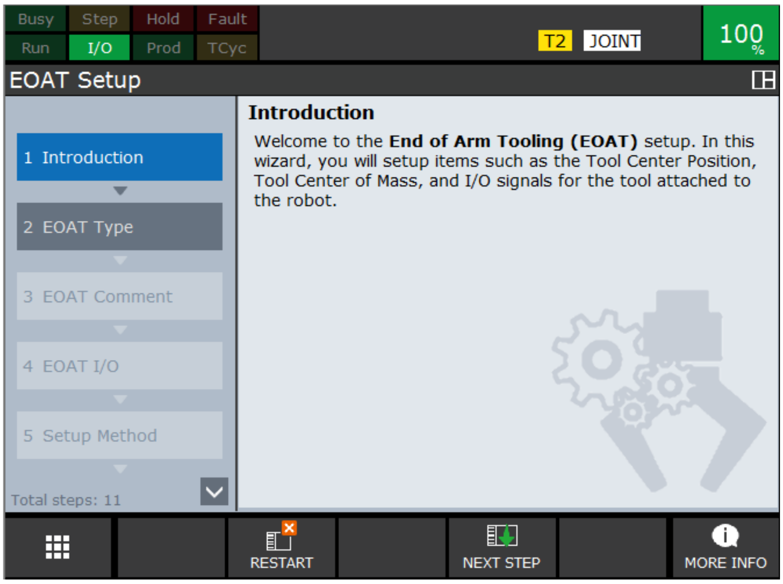


Fig. 12.2.2 End of Arm Tool Setup screen

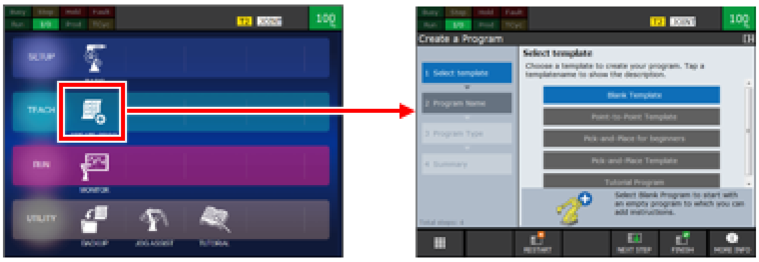
The main setup items on the EOAT Setup screen are as follows:
- EOAT Type
- EOAT Comment
- EOAT I/O
- Setup Method
- Straight or Offset
- Measurements
- Weight

12.3 OPERATIONS RELATED TO TEACH

Tap the [Create Program] icon on the home screen and display the Create a Program screen.
If there are multiple programs that have already been created, the [Select Program] icon will be displayed. An icon with the name of the program that is currently selected (example: PICK_PLACE) will also be displayed.

Fig. 12.3 (a) Teach icons

For program creation, refer to "12.3.1 Create Program." For information on when selecting existing programs, refer to "12.3.2 Select Program."
The screen display flow when there are no programs is shown below.



Home screen Create a Program screen

Fig. 12.3 (b) Screen display flow (when there are no programs)

12.3.1 Create Program

On the Create a Program screen, you create programs using template programs (sample programs) for achieving certain objectives. You can also select a "Blank Template" for creating a program from scratch.
The Create a Program screen is shown below. For the basic operations on the guide screens, refer to "12.1.2 Basic operations on the guide screens."

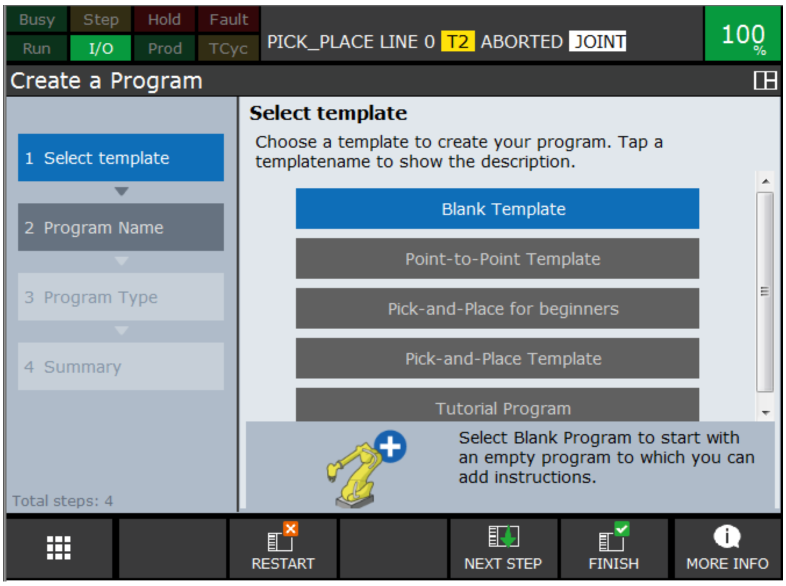


Fig. 12.3.1 Create a Program screen

The templates that can be selected on the Create a Program screen are as follows:

- Blank Template
- Point-to-Point Template
- Pick-and-Place for beginners
- Pick-and-Place Template
- Tutorial Program (This program will be used in "12.5.3 Tutorial")

The main setup items on the Create a Program screen are as follows:

- Select Template
- Program Name
- Program Type (when "Blank Template" is selected)

NOTE

If you select "Pick-and-Place for beginners" as the template, a tutorial is included in the subsequent flow, and the operation steps will be different from when other templates are selected.

12.3.2 Select Program

When a program has been created and is selected, an icon with the name of that program (example: PICK_PLACE) will be displayed. Also, if multiple programs have already been created, the [Select Program] icon will be displayed. If you tap this icon, the Select a Program screen will be displayed. An icon with the name of the program that is currently selected will be displayed in this case, too.

Fig. 12.3.2 (a) Teach icons

If you tap the target program name on the Select a Program screen, the Edit a Program screen will be displayed.
If you tap [All Programs] in the function key area on the Select a Program screen, a program list screen will be displayed.
The screen display flow when there are multiple programs is shown below.

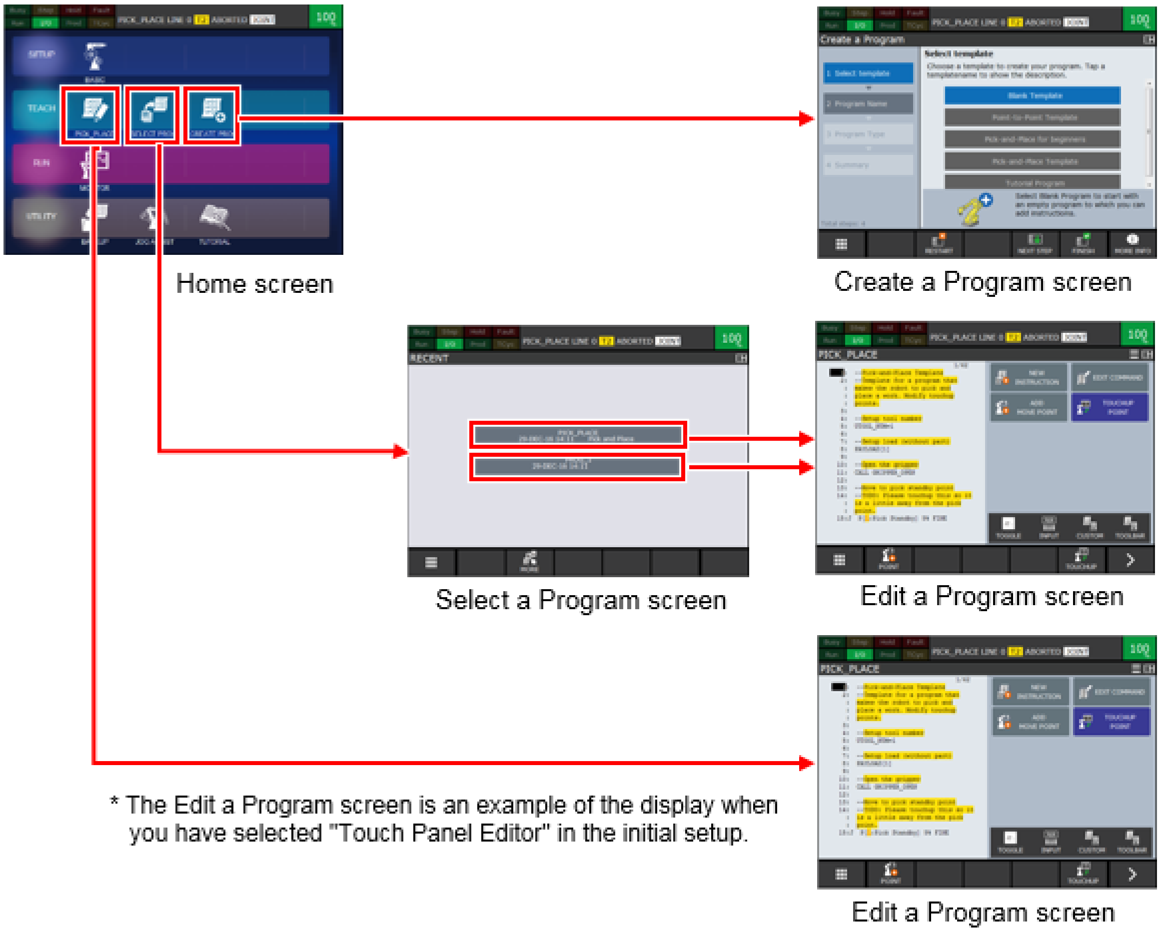


Fig. 12.3.2 (b) Screen display flow (when there are multiple programs)

12.4 OPERATIONS RELATED TO RUN

Tap the [Monitor] icon on the home screen and display the Production Monitor screen.
The screen display flow for execution is shown below.

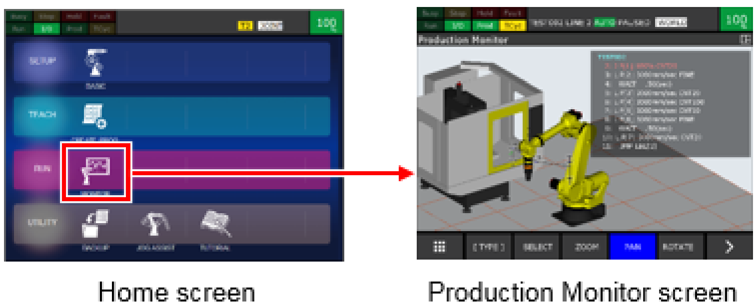


Fig. 12.4 Screen display flow (Run)

The Production Monitor screen displays the robot execution status in a graphical manner. Furthermore, the programs that are currently being executed and the teaching points can be checked.

12.4.1 Production Monitor

The Production Monitor screen is shown below.

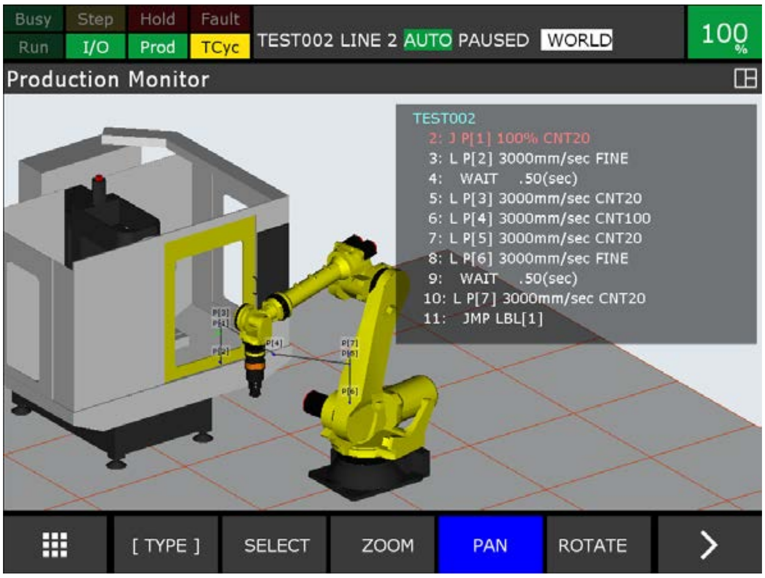


Fig. 12.4.1 Production Monitor screen

The Production Monitor screen is not in a step guide format like other guide screens, but is operated by means of function keys.

> **NOTE**
> In order to display peripheral devices and hands on the screen, you will need the "R764 4D graphic function" option.

Function key operations

Table 12.4.1 Production Monitor screen function keys

Function key	Description
[TYPE]	Switches the screen display. The screens that can be selected are the 4D graphic screen and the current position screen.
SELECT	This is an optional function.
ZOOM	Changes the magnification of the view. Increasing the magnification makes the target model larger, but the field of view will become narrower.
PAN	Moves the view up, down, left and right without changing the angle.
ROTATE	Changes the angle of the view without changing the position.
>	Switches the function key.
[VIEWS]	Changes the view (preset view, user view). For the user view, up to 8 views of your choice can be recorded. If you record a view, you can move to the recorded view instantly. You can also give a recorded view a name.
[VISIBLE]	This is an optional function.
CALLED	This is an optional function.

For the screen view operations and the 4D graphic display screen, refer to "7.7 4D graphic." For the current position screen, refer to "7.7.2 Current position."

12.5 OPERATIONS RELATED TO UTILITY

Tap the [Backup], [Jog Assist] or [Tutorial] icon on the home screen. [Backup] and [Jog Assist] display the Backup screen and Jog Assist screen respectively.
If you tap "Tutorial," the Tutorial Selection screen will be displayed. The Tutorial Selection screen displays the Tutorial: Jog a Robot screen, Tutorial: Program Edit screen, and Tutorial: Program Execution screen.

Fig. 12.5 Utility icons

For backups, refer to "12.5.1 Backup." For jog assist, refer to "12.5.2 Jog Assist." For utilities, refer to "12.5.3 Utility."

12.5.1 Backup

On the File Backup screen, you select devices to which backups will be made, select files and data to backup, and execute backups.
To display the File Backup screen, tap the [Backup] icon on the home screen.
The screen display flow for the File Backup screen and the File Backup screen are shown below. For the basic operations on the guide screens, refer to "12.1.2 Basic operations on the guide screens."

Fig. 12.5.1 (a) Screen display flow (Backup)

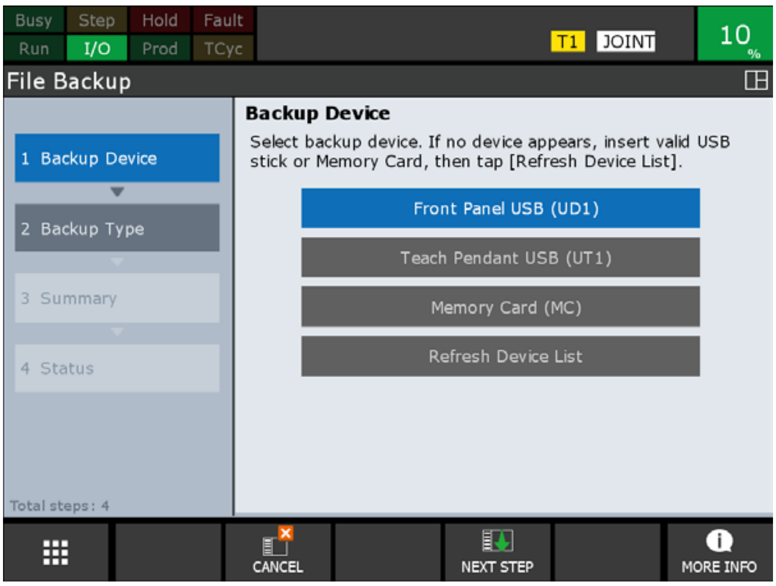


Fig. 12.5.1 (b) File Backup screen

The main setup items on the File Backup screen are as follows:

- Backup Device
- Backup Type

12.5.2 Jog Assist

On the Jog Assist screen, you set the override and coordinate system settings that are related to jog operation through icon selection operations.
To display the Jog Assist screen, tap the [Jog Assist] icon on the home screen.
The screen display flow for the Jog Assist screen and the Jog Assist screen are shown below.

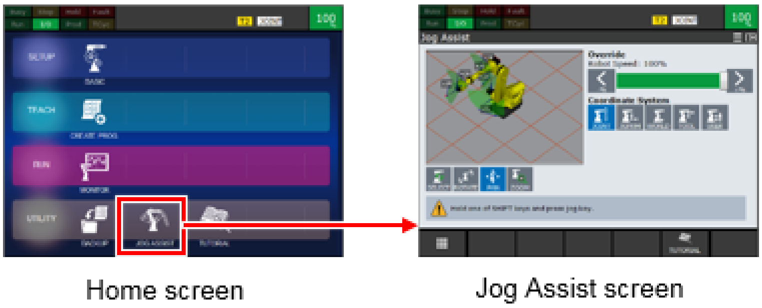


Fig. 12.5.2 (a) Screen display flow (Jog Assist)

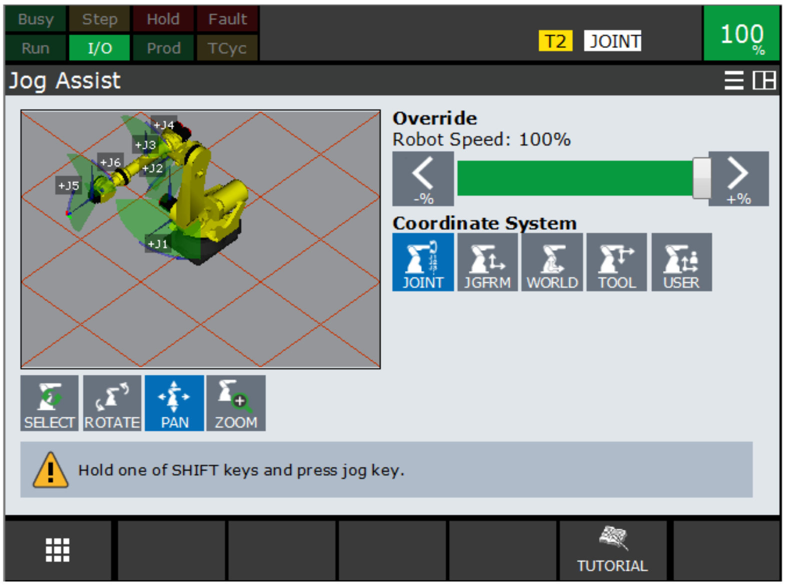


Fig. 12.5.2 (b) Jog Assist screen

The Jog Assist screen is not in a step guide format like other guide screens, but is operated by means of icons on the screen.

> ⚠ **WARNING**
> Before you jog the robot, be sure that all safety requirements for the work area are satisfied. Otherwise, injury or property damage could occur.

Override

Set the override value.
Tap < or >. The setting value changes in steps of 5%. You can also change the value by tapping the slide bar.

Coordinate system

Select the manual feed coordinate system. If you select jog, tool or user coordinate system, the [Select] key will be displayed in the function key area, and a list of coordinate system numbers that have been already set will be displayed. Select a coordinate system number from the list.

View operation

Table 12.5.2 View operation icons on the Jog Assist screen

Icon	Description
Rotate	Changes the angle of the view without changing the position.
Pan	Moves the view up, down, left and right without changing the angle.
Zoom	Changes the magnification of the view. Increasing the magnification makes the target model larger, but the field of view will become narrower.

If you tap [Tutorial] in the function key area after you have finished performing settings, the Tutorial: Jog a Robot screen will be displayed.

For the screen view operations, refer to "7.7 4D graphic."

12.5.3 Tutorial

In the tutorials, you can get experience of and learn about basic operations such as robot jog operation and program creation.
It is structured as shown below, so that you can get experience of and learn about the flow from "jog operation" to "program execution."

Table 12.5.3 Tutorial structure

Tutorial	Details
Jog a Robot	You will get experience of and learn about jog operation (joint / Cartesian) using the teach pendant.
Program Edit	You will get experience of and learn about how to create a program using the wizard and how to edit programs that you have created.
Program Execution	You will get experience of and learn about how to execute a program (single step operation / continuous test operation).

The screen display flow for tutorials is shown below.

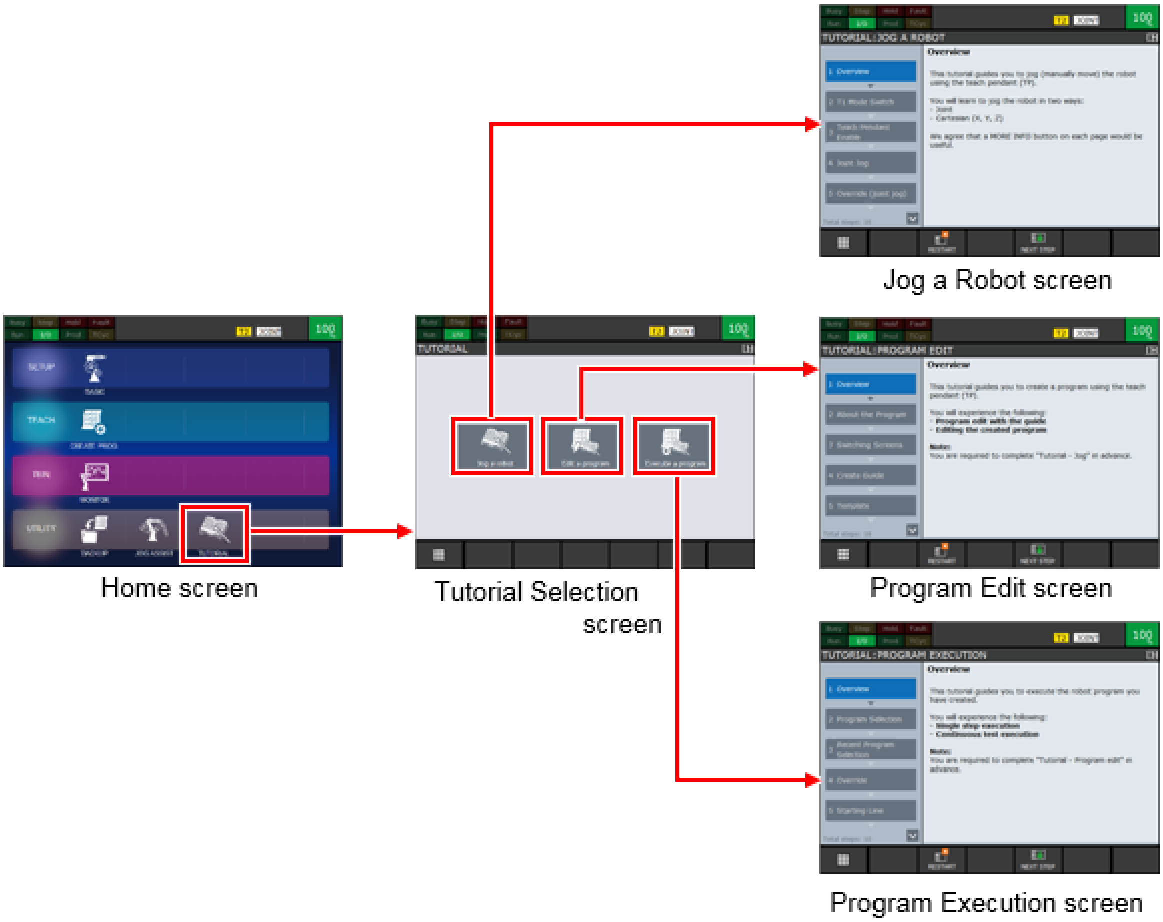


Fig. 12.5.3 Screen display flow (Tutorial)

12.5.3.1 Jog a Robot

On the Tutorial: Jog a Robot screen, you will get experience of robot jog operation using the teach pendant. You can move the robot by pressing the jog key on the teach pendant.
The Tutorial: Jog a Robot screen is shown below. For the basic operations, refer to "12.1.2 Basic operations on the guide screens."

Fig. 12.5.3.1 Tutorial: Jog a Robot screen

The jog operations you can get experience of on the Tutorial: Jog a Robot screen are joint jog and Cartesian jog (X, Y, Z). The main steps on the Tutorial: Jog a Robot screen are as follows:

- T1 Mode Switch
- Teach Pendant Enable
- Joint Jog
- Override (joint jog)
- DEADMAN switch
- Alarm Reset
- Joint jog (J1)
- Joint jog (J2 - J6)
- XYZ Jog
- Override (XYZ jog)
- X Direction
- Y/Z Direction
- Rotation Around X
- Rotation Around Y/Z

> ⚠**WARNING**
> Before you jog the robot, be sure that all safety requirements for the work area are satisfied. Otherwise, injury or property damage could occur.

12.5.3.2 Program Edit

On the Tutorial: Program Edit screen, you will get experience of program creation and editing using the teach pendant.

> **NOTE**
> In this tutorial, you use jog operation while editing a program. Therefore, it is recommended that you run "Tutorial: Jog a Robot" beforehand.

The Tutorial: Program Edit screen is shown below. For the basic operations, refer to "12.1.2 Basic operations on the guide screens."

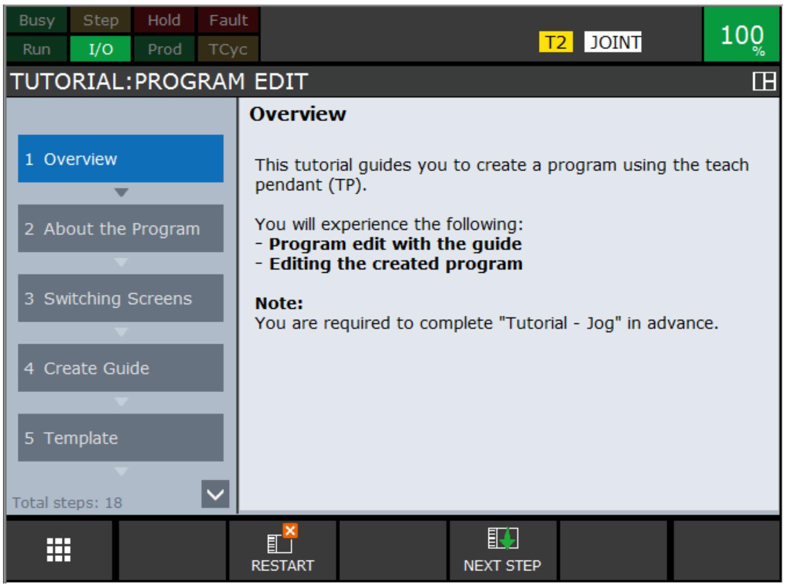


Fig. 12.5.3.2 Tutorial: Program Edit screen

The main steps on the Tutorial: Program Edit screen are as follows:

- About the Program
- Switching Screens
- Create Guide
- Template
- Summary Check
- Finish Program Create
- Program edit
- Home Position
- Move forward
- Move to side
- Move back
- Back to Tutorial

12.5.3.3 Program Execution

On the Tutorial: Program Execution screen, you will get experience of executing step operation and continuous test operation of the robot program that you have created.

NOTE
In order to run this tutorial, you need to have finished the "Tutorial: Program Edit" beforehand.

The Tutorial: Program Execution screen is shown below. For the basic operations, refer to "12.1.2 Basic operations on the guide screens."

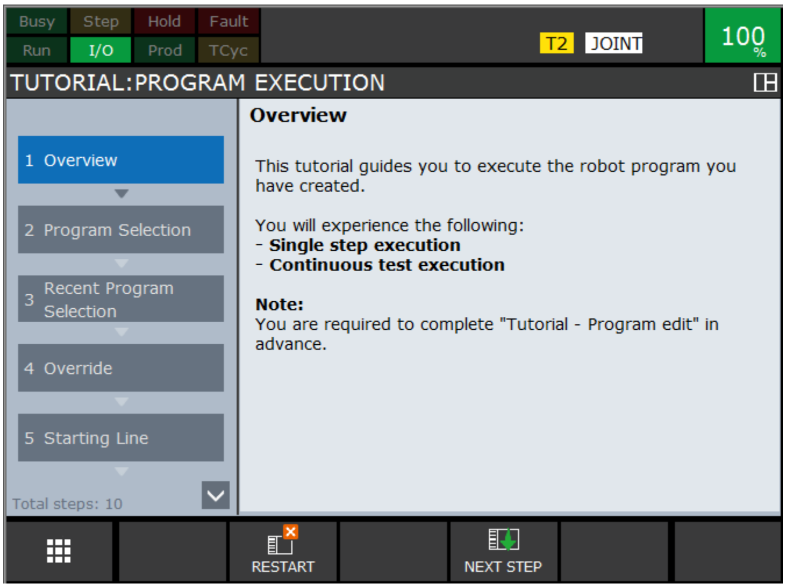


Fig. 12.5.3.3 Tutorial: Program Execution screen

The main steps on the Tutorial: Program Execution screen are as follows:

- Program Selection
- Recent Program Selection
- Override
- Starting Line
- Single Step
- Step Execution
- Continuous Mode
- Continuous Execution

13 OPERATIONS WITHOUT A TEACH PENDANT

This chapter describes operations when the system does not have a Teach Pendant.
For the R-30*i*B Compact Plus controller, the Teach Pendant is optional. A PC device can be used to teach robot programs instead of a Teach Pendant. Or, tablet device can be used.

Contents of this chapter

13.1 OVERVIEW

To operate the robot controller on a PC, *i*RProgrammer or the Remote *i*Pendant function should be used. For R-30*i*B Compact Plus these functions are provided by default, but other type of controllers need Software option.

Table 13.1(a) Software options required for each controllers

Controller	*i*RProgrammer	Remote *i*Pendant
R-30*i*B	Unavailable	Remote *i*Pendant option (R843)
R-30*i*B Mate	Unavailable	Remote *i*Pendant option (R843)
R-30*i*B Plus	*i*RProgrammer option (J767)	Remote *i*Pendant option (R843)
R-30*i*B Mate Plus	*i*RProgrammer option (J767)	Remote *i*Pendant option (R843)
R-30*i*B Compact Plus	Available by default	Available by default
R-30*i*B Mini Plus	Available by default	Remote *i*Pendant option (R843)

*i*RProgrammer has the following features.
- This function is available for use on common external device like a PC.
- The program editor is provided to create robot programs on a PC.
- Robot movement operation such as Jogging and TP program execution is allowed.
- One-touch Tree menu is provided on the PC.

*i*RProgrammer is supported on the R-30*i*B Compact Plus, the R-30*i*B Plus, the R-30*i*B Mate Plus and the R-30*i*B Mini Plus. Software option (J767) is needed to use the *i*RProgrammer function for the R-30*i*B Plus and the R-30*i*B Mate Plus.

Remote *i*Pendant has the following features.
- This function is available for use on common external device like a PC.
- The following modes of the Remote *i*Pendant can be used for teaching and operation of the robot systems.

  Jogging *i*Pendant (JITP)

    An *i*Pendant-like screen is provided on the PC.

    Robot movement operation such as Jogging and TP program execution is allowed.

Navigate *i*Pendant (JCGTP)
An *i*Pendant-like screen is provided on the PC.
Robot movement operation such as Jogging and TP program execution is not permitted.

These functions work on a web browser. Installing of any additional software is not required.
The following browsers are supported.
- Google Chrome (after version 31)
- Internet Explorer 11

> ⚠ **WARNING**
> - When the Robot controller does not have a Teach Pendant, the robot must be operated from outside of the safeguarded zone in AUTO mode. Otherwise, serious personal injury could result.
> - When the robot is moved by a PC, an emergency stop button must be connected and located near the operator for use at any time.

13.2 SETUP

To connect to your PC using the *i*RProgrammer and the Remote *i*Pendant function, you must set the TCP/IP setting on the robot controller.

The IP address of the controller is set by the default for the R30*i*B Compact Plus as following.
Port#1 IP address: 192.168.1.100
Subnet Mask: 255.255.0.0
Router IP address: 192.168.1.1
PC JOG IP address: *
You will need to set the IP address of your PC to access the controller.
For Example:
IP address: 192.168.1.200
Subnet Mask: 255.255.0.0
When Ethernet port1 of the robot controller is connected to the PC directly by means of an Ethernet cable, *i*RProgrammer and the Remote *i*Pendant function can be used without changing any controller settings.

For the R-30*i*B Plus and the R-30*i*B Mate Plus, initial setting of Ethernet is not done.
In order to connect to the PC, the IP address of the controller and the PC should be set up.

If the PC JOG IP address is set, the PC which can access to the controller is restricted by *i*RProgrammer and Jogging *i*Pendant. Only the device that has the same IP address as “PC JOG IP address” can access the controller. For the R-30*i*B Comact Plus, the default value of the “PC JOG IP Address” is*. This means that any devise can connect to the controller using *i*RProgrammer and Jogging *i*Pendant. For the R-30*i*B Plus and the R-30*i*B Mate Plus, the default value of the “PC JOG IP Address” is 192.168.0.100. “PC JOG IP Address” should be changed according to the IP address of device to be connected.
To change a network setting without *i*Pendant, please refer to “13.3.5.3 Configure Ethernet”.
To find out more about TCP/IP settings, please refer to the chapter “2 SETTING UP TCP/IP” in the “Ethernet Function OPERATOR’S MANUAL(B-82974EN)” for details.

13.3 OPERATIONS

13.3.1 Connect to Controller

For the R-30*i*B Compact Plus, when the controller is powering up, the CYCLE START LED on the controller panel blinks.
When the blinking stops, the startup process is completed. The PC can then connect to the controller using *i*RProgrammer or Remote *i*Pendant.

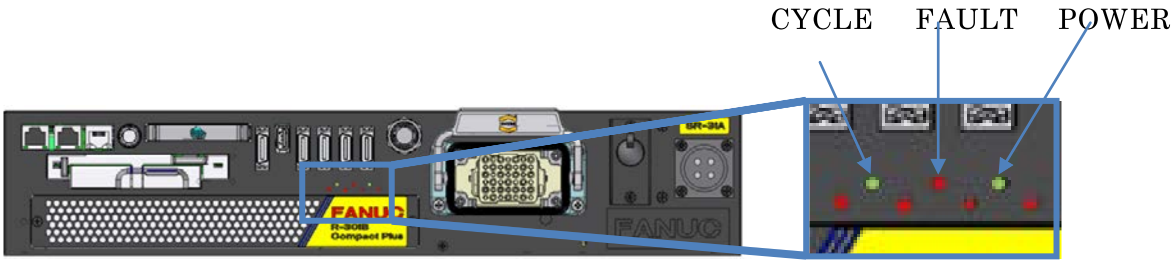


Fig. 13.3.1(a) LED on R-30*i*B Compact Plus Controller

*i*RProgrammer and the Remote *i*Pendant function can be started from the robot Homepage. Some functions can be selected from links in the page.

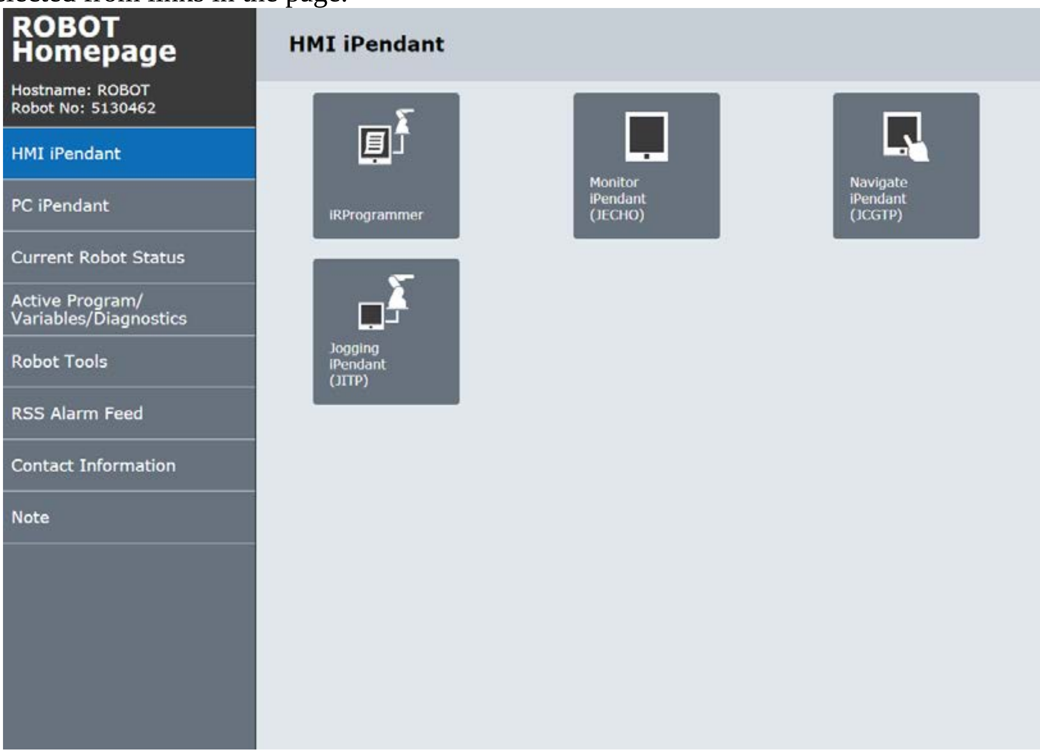


Fig. 13.3.1(b) Robot homepage screen

Table 13.3.1(a) Items in Robot homepage HMI *i*Pendant

Item	Description
*i*RProgrammer	Open *i*RProgrammer on the browser. Please refer “13.3.2 *i*RProgrammer”.
Monitor *i*Pendant (ECHO)	Open Monitor *i*Pendant on the browser. This function provides you with the capability to display and monitor the current *i*Pendant screens. This function is not used when there is no *i*Pendant. Please refer to the chapter “Remote *i*Pendant” in the “Ethernet Function OPERATOR’S MANUAL(B-82974EN)”
Navigate *i*Pendant (JCGTP)	Open Navigate *i*Pendant on the browser. Please refer “13.3.4 Navigate *i*Pendant.
Jogging *i*Pendant (JITP)	Open Jogging *i*Pendant on the browser. Please refer “13.3.3 Jogging *i*Pendant.

Procedure 13-1 Connection to *i*RProgrammer and Remote *i*Pendant

Conditions

One of the following conditions has to been satisfied.

1 The controller is a R-30*i*B Compact Controller.
The network setting of the controller is the default setting.
The IP address of your PC is set to access the controller.
Ethernet port 1 of the controller is connected directly to the PC by an Ethernet cable.

2 The PC and the robot controller are connected to a network.
Their IP addresses are set to conform to the network.
Please refer “13.3.5.3 Configure Ethernet” for information on setting up the network without an *i*Pendant.

Steps

1 Activate your browser on your PC.

2 In the Address field of the browser, input “http://<Robot IP Address>” to open the robot Homepage (Example: “http://192.168.1.100” for the default system). If the connection is successfully made you will briefly see the robot homepage, similar to that shown in Fig. 13.3.1(b), displayed on the PC.

3 Select the link you want to use from the HMI *i*Pendant section.
For *i*RProgrammer and Jogging *i*Pendant, if an actual *i*Pendant is connected, the following operations are needed before there functions are started.
- Turn off the TP enable switch
- Select AUTO mode

When *i*RProgrammer or Joggin *i*Pendant is opened, any actual *i*Pendant it will be disconnected automatically and you will see an *i*Pendant screen as shown in Fig. 13.3.1(c).

Fig. 13.3.1(c) Disconnected *i*Pendant screen

NOTE
Navigate to the *i*Pendant (JCGTP) function using the standard port No.80.
Jogging *i*Pendant (JITP) and *i*RProgrammer use special ports 3080 and 4080.

Procedure 13-2 Disconnection from *i*RProgrammer and Remote *i*Pendant

To disconnect the *i*RProgrammer and Remote *i*Pendant function, press the “Logout” key at the bottom of the screen before closing the window of the browser. Some devices do not close the network connections properly if you close the window without logging out.

13.3.2 *i*RProgrammer

*i*RProgrammer is used to create and edit programs.
*i*RProgrammer provides the following feature.

- Programing Editor
  1. Text based editor to edit programs
  2. A tabular form to edit position data
  3. Jog operation using a Jog Panel
  4. Ability to rum programs from the PC
- Setting screens
  1. Basic setting screens (Frames, Payload, etc.)
  2. Screens to monitor robot controller status (I/O, Error, etc.)

⚠ **WARNING**
*i*Rprogrammer enables you to move the robot using a PC. Extreme caution must be exercised when using this function. When you execute jog operation using *i*Rprogrammer, you must strictly adhere to the following precautions:
- Robot operation must be executed from outside of the safeguarded space.
- You must check to ensure that no one is inside of the safety fence.
- The Emergency stop button must be connected and located near the operator for use at any time.
- Robot operation must be executed with the robot in sight.

13.3.2.1 Screen

Use Procedure 13-1 to open *i*RProgrammer by selecting the link of the “*i*RProgrammer” in the robot homepage.
When the connection is successfully made you will briefly see the screen, similar to that shown in Fig. 13.3.2.1, displayed on the browser on your PC.

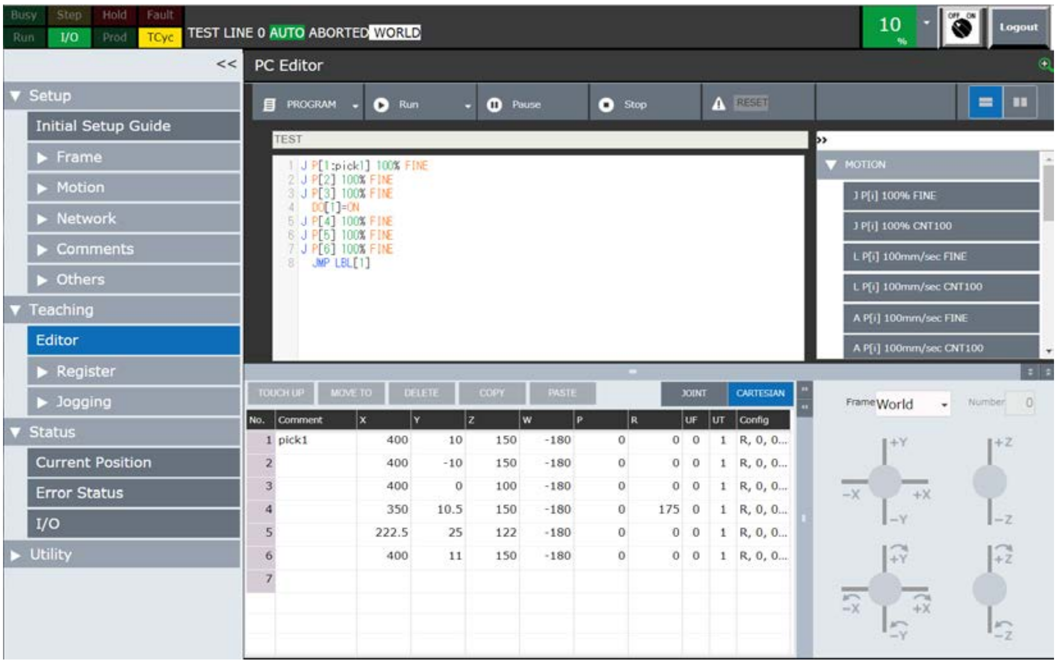
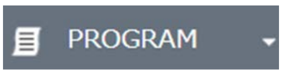


Fig. 13.3.2.1(a) *i*RProgrammer

Table 13.3.2.1(a) Element in *i*RProgrammer Editor

Item	Description
Menu	The tree-based menu lists menus to edit programs and basic settings. Open each screen by clicking or touching the item.
Programing text Editor	Text based editor to edit program instructions. To input an instruction, use a keyboard or select the item from the instruction list.
Instruction List	The instruction list is used for robot programming. When an item is selected, the item is added to the cursor position in the editor.
Position Table	The Position Table is used to edit positional data. Position is taught by inputting the value from a keyboard or by using the TOUCHUP key.
Jog Pane	Jog buttons used to jog the robot.
Tool bar	The Tool bar displays items used to create programs and execute programs. Please refer Table 13.3.2.1(b).
Status bar	The Status bar has status LEDs and an area to display error messages. Please refer Table 13.3.2.1(c).

Table 13.3.2.1(b) Keys in Tool bar

Keys	Description
 PROGRAM	The following items are in this menu. Open: Select programs to edit. Create Prog: A new program is created. Save: The edited program is saved. The program being edited is written into temporary memory before the save operation. To start running the program, it has to be saved. Save As: The edited program is saved as different name. Rename: Change selected program name.
 Run	Start running the program specified on the status bar. The TP enable switch on status bar has to be ONto start running the program. When the arrow button is clicked, running mode can be selected; Single Step or Continuous. When Single Step is selected, Step LED on the status bar turns ON.

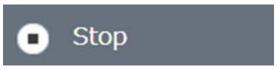
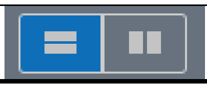
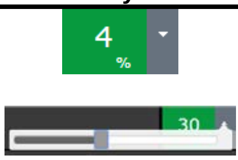
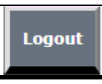
Keys	Description
	Pause the running program.
	Aborts the program.
	Reset alarm. When any alarm occurs, this button is changes to red.
	Changes to split screen mode.

Table 13.3.2.1(c) Keys in Status bar

Keys	Description
	Slider to change override value. The slider is displayed when an arrow button is clicked.
	The TP enable switch allows you to move the robot using *i*RProgarmmer. When the key is ON, the robot can be moved by jogging and program execution.
	The Logout key to correctly logout from the *i*RProgrammer screen.

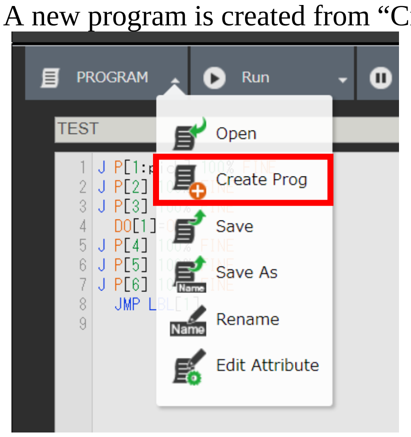
13.3.2.2 Program Editing

*i*RProgrammer provides the following programing operations. Programs can be edited in the editor Screen.

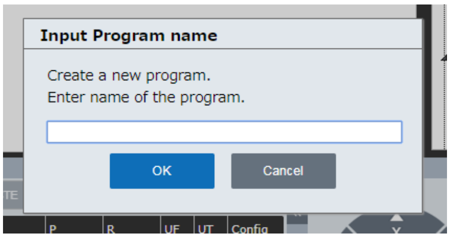
- Create new programs
- Texted based Programing using keyboard
- Select instructions from the tree list
- Edit position data using tabular form

Create Program

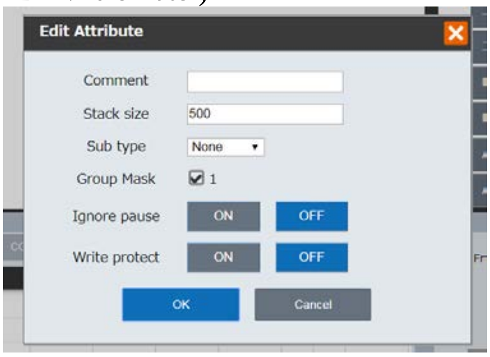
A new program is created from "Crate Prog" item in PROGRAM menu on the tool bar.



A dialog box to input the program name is displayed. Input new program name.

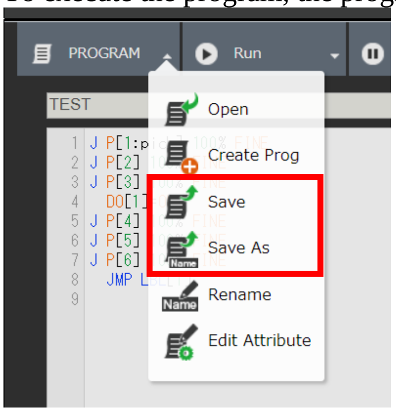


A dialog box to edit attribute is displayed. Edit attribute of new program. (Supported in software version 7DF1/10 or later)



Save Program

The Editing program is saved as a temporary image. Please save programs after editing is completed.
When the SAVE item in PROGRAM menu is selected, the program is saved as an actual program.
To execute the program, the program has to be saved.



Select Program

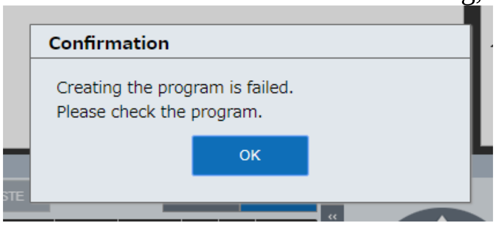
Existing program can be selected from the program selection box.
When the arrow key of the selection box is pushed, the programming list is displayed.
When a program is selected the program is displayed in the text editor and position table.

Edit instruction

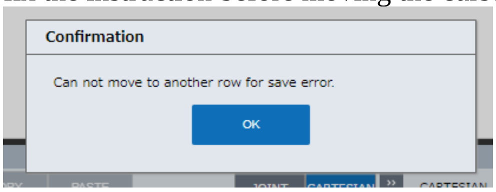
Program instructions are edited in the text based editor.
Instructions can be input using a keyboard or selected from the instruction list.
When user input characters are input using a keyboard, possible instructions are can be chosen to be added.

When an item in the instruction list is clicked, a dialog to input detail setting for the selected instruction is displayed. The selected instruction is added at the position of the cursor is in the program.

When an instruction in the line is wrong, an error occurs.



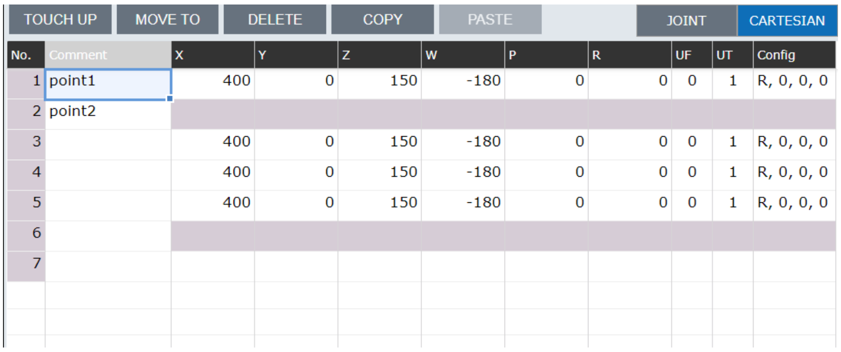
When an error occurs on the line, the cursor cannot be moved to other line with the following error. Please fix the instruction before moving the cursor.



> **NOTE**
> If software version 7DF1/11 or former, input ";" on the end of each line. Error occurs when ";" is not on the line.

<u>Edit Position data</u>

Position data is edited in tabular form.
A position is taught by inputting a value from a keyboard or by using the TOUGHUP key.



No.	Comment	X	Y	Z	W	P	R	UF	UT	Config
1	point1	400	0	150	-180	0	0	0	1	R, 0, 0, 0
2	point2									
3		400	0	150	-180	0	0	0	1	R, 0, 0, 0
4		400	0	150	-180	0	0	0	1	R, 0, 0, 0
5		400	0	150	-180	0	0	0	1	R, 0, 0, 0
6										
7										

Fig. 13.3.2.2(a) Position table

Table 13.3.2.2(a) Items in Position table

Item	Description
TOUCH UP	Teach current position to the position on the cursor line.
MOVE TO	The robot moves to the position on the cursor line. The TP enable key has to be ON to move the robot. (Supported in software version 7DF1/15 or lator)
DELETE	Delete position data on the cursor line. If the position is used in motion instruction, the position data cannot be deleted.
COPY	Copy position data in a line or value in a cell
PASTE	Paste copied position data or value.
JOINT/CARTESIAN	Select table type, Joint representation or Cartesian representation. Position data cannot be edited, if the position data is set using a different representation type.

13.3.2.3 Jog operation

The TP enable key has to be ON when you jog the robot using the JOG Panel. When a key in the JOG panel is pressed, the robot moves only while the key is being pressed.
The coordinate system used to jog operation is selectable from drop-down list that has Joint, World, Tool, and User coordinate system.

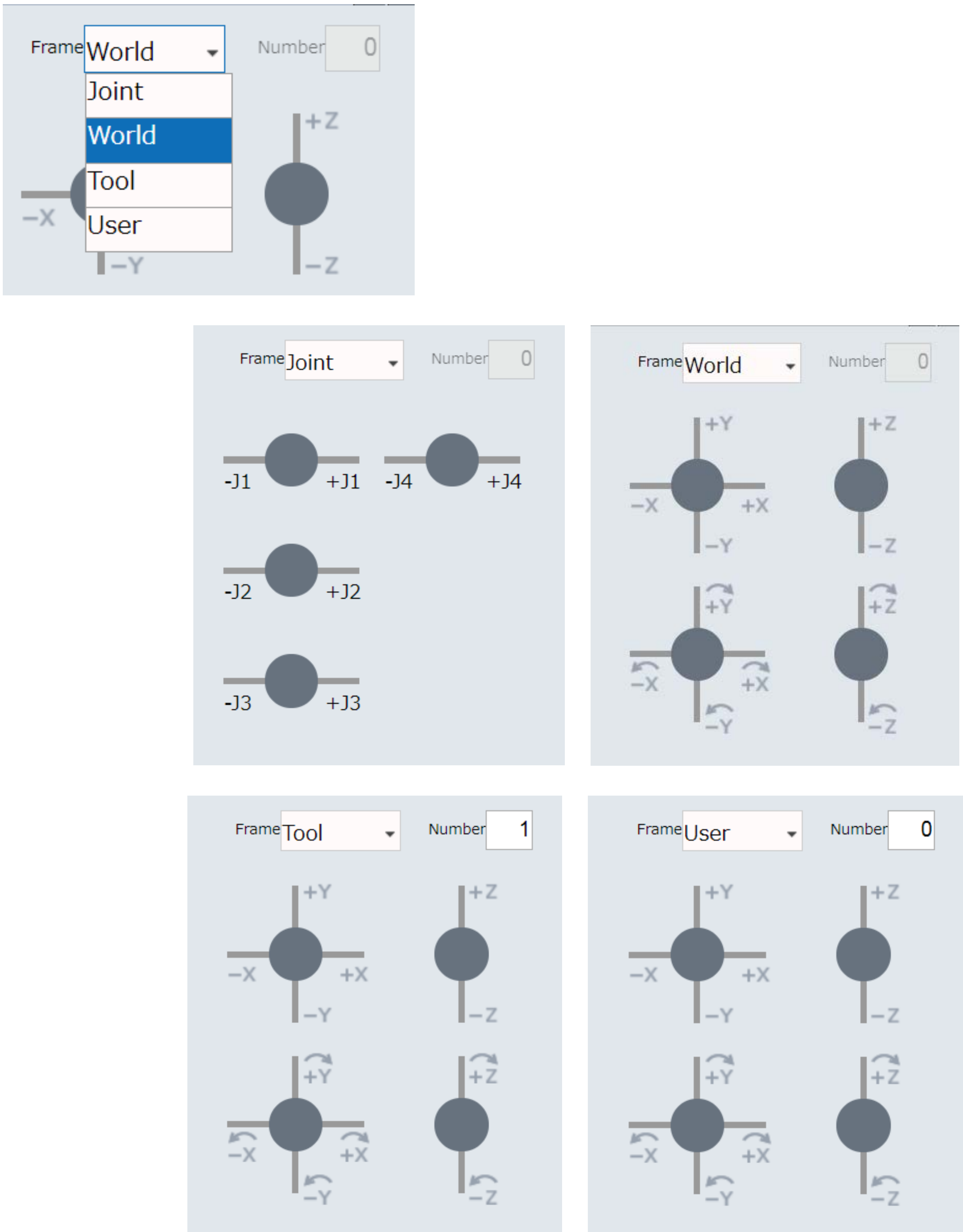


Fig. 13.3.2.3(a) Jog Panel

⚠ WARNING

When you jog the robot using the Jog Panel, you must strictly adhere to the following precautions.

- Jog operation must be executed from outside of the safeguarded space.
- You must check that no one is inside the safety fence.
- An Emergency stop button must be connected and located near the operator for use at any time.
- Jog operation must be executed with the robot in sight.

13.3.2.4 Program execution

The TP enable key has to be ON to run programs. When the RUN key on the tool bar is pressed, the program is executed.
When the arrow key is pressed, the running mode can be selected; Single Step or Continuity. When Single Step is selected, the Step LED on the status bar is turned ON.

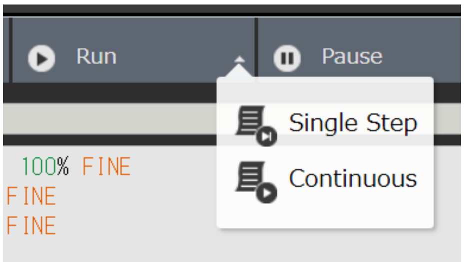
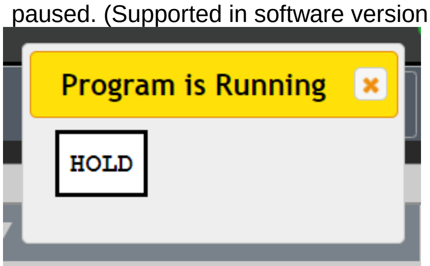


Fig 13.3.2.4(a) Single Step or Continuous

“Program is Running” dialog is displayed during program is running. Even if leave program editor and move to other screen, “Program is Running” dialog is displayed if program is running. If push the Hold key in the dialog, program is paused. (Supported in software version 7DF1/12 or later)



⚠ WARNING

Verify the selected program and line number is correct before running a program .

⚠ WARNING

When you start running programs using *i*RProgrammer, you must strictly adhere to the following precautions.

- Running programs must be executed from outside of the safeguarded space.
- You must check that no one is inside in the safety fence.
- An Emergency stop button must be connected and located near the operator for use at any time.
- Jog operation must be executed with the robot in sight.

> ⚠ **WARNING**
> When programs are executed as using *i*RProgrammer , the executed program may be different from a displayed program. Be sure to save any edited programs in *i*RProgrammer before programs are executed.

13.3.2.5 Tree menu

Tree menu has following items.

Table 13.3.2.5(a) Tree Menu

Section	Menu1	Menu2	Remark
Setup	Initial Setup Guide		Initial setup
	Frame	Utool	Tool Frame setting
		Uframe	User Frame setting
	Motion	Payload	Payload setting
		Reference Position	Reference Position setting
		Axis Limit	Axis Limit setting
	Network	Host Comm	Network setting
	Comments	String Register	String Register comment setting
		User Alarm	User alarm comment setting
	Others	Program Setup	Setting for program execution
		General Setup	General setting (language etc.)
		DCS	DCS setting
Teaching	Editor		Programming Editor
	Register	Numeric Register	Show and set numeric register
		Position Register	Show and set or record position register
	Jogging	Touch Jog	Jogging by using Touch Jog screen
		Jog Panel	Jogging by using Jog Key
Status	Current Position		Display current position
	Error Status		Display error status
	I/O		Show and set I/O value
Utility	File		Display files in devices
	File Backup Guide		Backup Guide
	Cycle Power		Select start mode for next startup

13.3.2.6 Limitations of *i*RProgrammer

The following limitations apply to the *i*RProgrammer function.

- *i*RProgrammer connection is refused when the robot mode in T1 or T2.
- Multiple *i*RProgrammer connection is not supported.
- Teaching Pendant cannot be operated when *i*RProgrammer is connected.
- *i*RProgrammer connection is refused when Teaching Pendant is started to connect.
- Screen size of device is recommended more than 9 inch to use *i*RProgrammer.
- Teaching Pendant has some functions that are not displayed on tree menu.
- There are unsupported instructions in programing editor. Point logic, palletizing, tracking are unsupported.
- Supported languages are English, Japanese and Chinese. The screen is displayed at English when other language is selected.
- The program cannot be run from arbitrary line. The program starts from first line, or the program restarts from paused line.
- The program that started by *i*RProgrammer stops when *i*RProgrammer is disconnected.

13.3.3 Jogging *i*Pendant (JITP)

This term describes about basic functions of Jogging *i*Pendant. Please refer to the chapter “16 REMOTE *i*Pendant FUNCTION” in the “Ethernet Function OPERATOR’S MANUAL(B-82974EN)” for details.

The Jogging *i*Pendant function allows you to display an *i*Pendant-like screen using a browser on a PC. This function provides *i*Pendant screens independent of the actual *i*Pendant. It also provides robot movement operation such as Jogging and TP program execution.

⚠ **WARNING**

The Jogging *i*Pendant function enables you to move the robot using a PC. Extreme caution must be exercised when using this function. When you jog using the Jogging *i*Pendant function, you must strictly adhere to the following precautions.

- Set the Mode Switch to AUTO mode.
- Jog operation must be executed from outside of the safety fence.
- You must check that no one is inside the safety fence.
- Emergency stop button must be connected and located near the operator for use at any time.
- Jog operation must be executed with the robot in sight.

13.3.3.1 Connection

Use Procedure 13-1 to open the Jogging *i*Pendant by selecting the “Jogging *i*Pendant(JITP)” link, in the robot homepage.

When the connection is successfully made you will briefly see a screen, similar to that shown in Fig. 13.3.3.1, displayed on the browser on your PC.

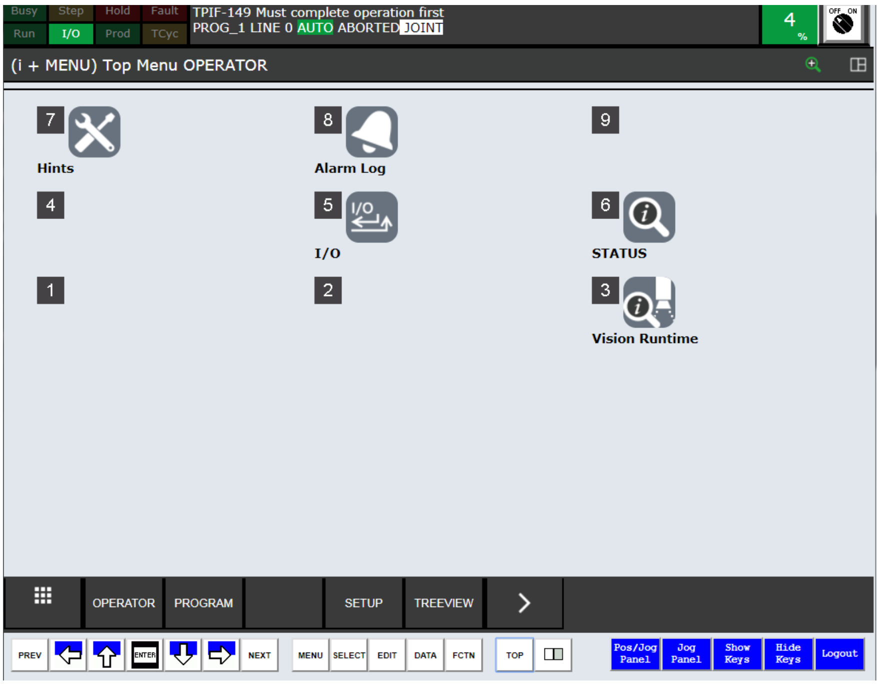


Fig. 13.3.3.1(a) Jogging *i*Pendant screen

13.3.3.2 Screens

The Jogging *i*Pendant screen has the auxiliary keys available at the bottom of the Jogging *i*Pendant screen. The keys for the Jogging *i*Pendant function are described in Table 13.3.3.2.

Table 13.3.3.2(a) Jogging *i*Pendant (JITP) keys

Keys	Description
PREV	This key moves to Previous Screen or Cancels a pending operation.
← ↑ ↓ →	The cursor key moves the cursor in the direction of the arrow.
ENTER	The ENTER key enters a numeral or selects a menu.
F→	This key to switches the function key menu to the next page.
MENU	The MENU key to displays the menu screen.
SELECT	The SELECT key to displays the program selection screen.
EDIT	The EDIT key displays the program edit screen.
DATA	The DATA key to displays the program data screen.
FCTN	The FCTN key to displays the function menu.
TOP	The TOP key to displays the Top Menu OPERATOR screen.
	In case the screen on the teach pendant is split, when this key is pressed, the operation target screen is changed.
DISP	When this key is pressed, the menu to split the screen is displayed.
Show Keys	When this key is pressed, teach pendant keys are displayed to the right side of the screen.
Hide Keys	When this key is pressed, keys displayed to the right side of the screen hide.
Logout	The Logout key logs out of the Navigate iPendant (JCGTP) screen.
Pos/Jog Panel	When this key is pressed, position screen is displayed to the left side of the screen and jog keys are displayed to the right side of the screen.
Jog Panel	When this key is pressed, jog keys are displayed to the right side of the screen.
OFF ON	The TP enable switch key changes the ENABLE/DISABLE state of the teach pendant of the jogging *i*Pendant.

When you press "Show Keys" on lower part of the Jogging *i*Pendant screen, the key sheet shown in Fig. 13.3.3.2(a) will be displayed on the right side of the screen. Motion keys removed from the actual iPendant are located on the key sheet. "Hide keys" is pressed, keypad is disappeared and screen is displayed as single screen.

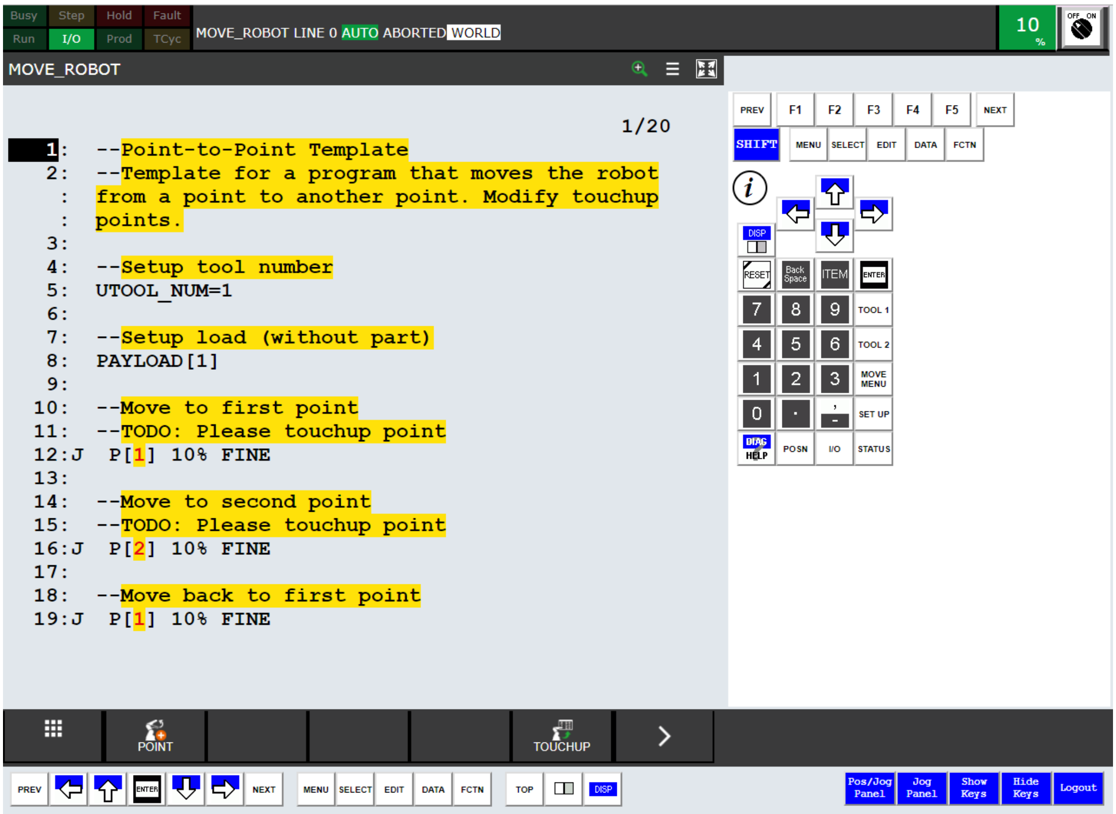


Fig. 13.3.3.2(a) Operation key sheet

When you press “Hide keys” on lower part of the Jogging *i*Pendant screen, the key sheet hides and the screen changes to single.

When you press Jog Panel”, Jog panel is displayed in the right hand side of the screen as shown in Fig. 13.3.3.2(b). Jog Panel consist of motion keys on the actual *i*Pendant key sheet.
When you press “Hide Keys”, the Jog Panel hides and the left screen goes full-screen.

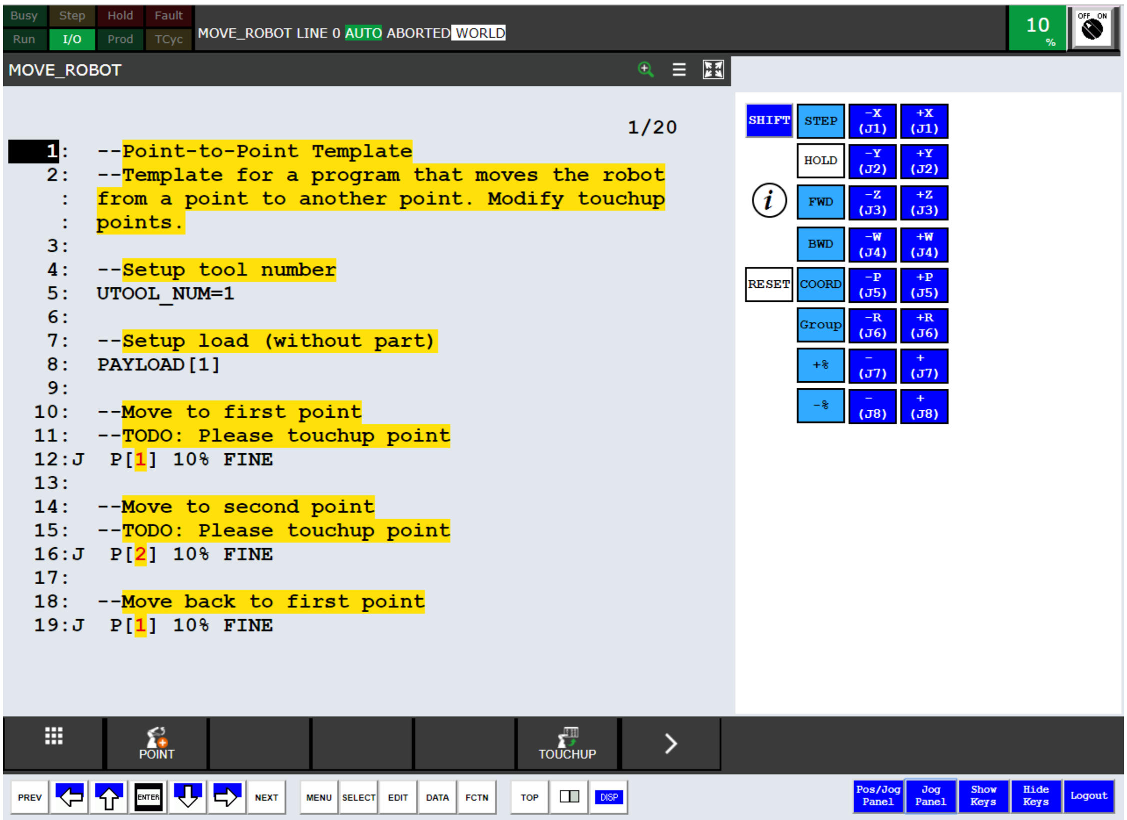


Fig. 13.3.3.2(b) Jog Panel screen

13.3.3.3 Jog operation

This subsection provides the sequence to jog the robot from Jogging *i*Pendant screen. All steps described in Procedure 13.3 can be performed on a PC.

Procedure 13-3 Robot Jog Operation

Steps

1 Show the Jog Panel by pressing the "Pos/Jog Panel", "4D/Jog Panel" or "Jog Panel" keys.

2 Press the TP enable/disable switch located in the upper right of the screen to turn it ON.

3 Press the RESET key to clear the "SYST-045 TP enabled in AUTO mode" alarm.

4 Press the SHIFT key.
When you press the SHIFT key, the color of the key is changed from blue to aqua. This means that the SHIFT key is pressed and in a hold state (latched state). When you press the SHIFT key again, the color turns to blue. This means the SHIFT key is released. Refer to Table 13.3.3.3(a) for details.

Table 13.3.3.3(a) SHIFT key state

Keys	State
SHIFT	Released SHIFT key.
SHIFT	Pressed and held SHIFT key.

5 When press and hold a jog key with latching SHIFT key, the robot moves.

⚠ WARNING

Jogging *i*Pendant function enables to move the robot by PC. Extreme caution must be exercised when using this function. When you jog operation, you must strictly adhere to the following precautions.

- Set the Mode Switch to AUTO mode.
- Jogging must be executed from outside of the safeguarded space.
- You must check the on one is inside the safety fence.
- An Emergency stop button must be connected and located near the operator to use any time.
- Jog operation must be executed with the robot in sight..

13.3.4 Navigate *i*Pendant (JCGTP)

This term describes about basic functions of Navigate *i*Pendant. Please refer to the chapter “16 REMOTE *i*Pendant FUNCTION” in the “Ethernet Function OPERATOR’S MANUAL(B-82974EN)” for details.

The Navigate *i*Pendant function allows you to display an *i*Pendant-like screen on a browser on your HMI device. Navigate *i*Pendant provides *i*Pendant screens independent of the actual *i*Pendant. So the PC device user can display screens, navigate menus, and enter data independent from an actual *i*Pendant.

⚠ CAUTION

Remote Operation is completely independent from the operation of the actual *i*Pendant therefore it can affect the operation the robot controller. Extreme caution must be exercised when using this feature.

13.3.4.1 Connection

Navigate *i*Pendant is opened when link of the “Navigate *i*Pendant(JCGTP)” link in the robot homepage is selected by Procedure 13-1.
When the connection is successfully made you will briefly see a screen, similar to that shown in Fig. 13.3.4.1, displayed on the browser on your PC.

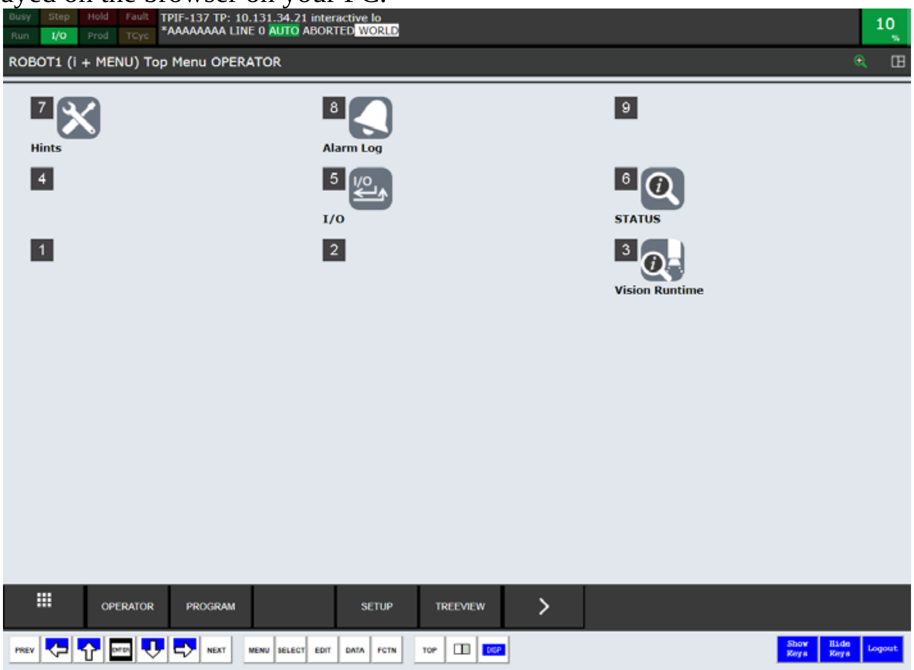


Fig. 13.3.4.1(a) Navigate *i*Pendant screen

13.3.4.2 Screens

While in the Navigate *i*Pendant connection, you can use a mouse and/or touch operation to select any of the available function keys or the auxiliary keys available below the *i*Pendant screen. These auxiliary keys are almost the same as Jogging *i*Pendant (Table 13.3.3.3), except that “POS/JOG Panel”, “JOG Pane” and “TP enable switch” are not displayed.

13.3.3 Special Operation

The R-30*i*B Compact Plus and R-30*i*B Mini Plus controllers can perform the following operations by means of the XML file, that is saved on the UD1: device, when power is turned on.

- Select start mode
- Configure Ethernet
- Set PC Jog IP address
- Image backup/Image restore

13.3.5.1 Operation

In order to perform any of the above operations, you have to write the tag in STARTUPCONFIG.XML that is saved on the UD1: device. Also, you have to get the key that allows execution of the XML in advance. Then write the key in STARTUPCONFIG.XML.

Whether you perform any operation, the following tag is always required. However, you have to change the key in the CONTROLLER tag to what was generated in BOOTKEY.DAT.

```
<?xml version="1.0" encoding="utf-8"?>
<ROOT>
  <CONTROLLER expires = "07/30/2016 - 19:25:34" key =
"590174ed-4b69388c ">

  </CONTROLLER>
</ROOT>
```

If STARTUPCONFIG.XML exists in UD1:, BOOTKEY.DAT will be created in UD1:.
In BOOTKEY.DAT, a new key will be generated. The expiration date of the key is one hour form the generation of the key.
If the key that written in STARTUPCONFIG.XML is incorrect or it is expired, the operation of XML will not execute.

When executing the operation, you have to write the operation tag between <CONTROLLER> and </CONTROLLER>. Usable tags are as follows.

Table 13.3.5.1(a) Tag

Tag	Description example of tag	Remarks
STARTMODE	- <STARTMODE value = "CTRL" /> - <STARTMODE value = "COLD" />	Set the start mode. You can select controlled start or cold start.
ETHERNET	- <ETHERNET id ='port1' label='CD38A' ip = "172.168.192.10" router = "172.168.1.1" mask = "255.255.240.0" hostname = "MY_ROBOT" /> - <ETHERNET id ='port1' label='CD38A' ip = "DHCP"/>	Change Ethernet settings such as IP address. Automatic setting is also possible by DHCP.

Tag	Description example of tag	Remarks
PENDANT	- <PENDANT ip = "172.168.192.200" />	Set IP address of the device that allows JITP connection.
CONFIG	- <CONFIG image = "UD1:\IMG\" startmod = "backup"/> - <CONFIG image = "UD1:\IMG\" startmod = "restore" dcsparam = "1" />	Execute image backup or image restore.

NOTE
If the XML command is incorrect, unintended operation may occur. Please check the XML command thoroughly before execution.

If XML command was successfully executed, STARTUPCONFIG.XML is renamed to OLDSTARTUPCONFIG.XML. If you want to run XML command again, you need recreate STARTUPCONFIG.XML.

NOTE
When you rename OLDSTARTUPCONFIG.XML to STARTUPCONFIG.XML, the XML command is executed if the key is within the expiration date. If you change the file name, please check the XML command before turning on the power.

Procedure 13-4 Operation by XML

Steps

1 Create STARTUPCONFIG.XML in the USB memory device and plug it into UD1:.
2 Boot the controller. BOOTKEY.DAT, EXAMPLE.XML, and STARTUPCONFIG.LOG are generated in the UD1: device.
In BOOTKEY.DAT, the key that allows execution of XML and CONTROLLER tags are generated.
In EXAMPLE.XML, a description example of STARTUPCONFIG.XML is generated.
In STARTUPCONFIG.LOG, the log used during execution of the XML is written. Error information (if execution fails) is also written here.
3 Describe the content of STARTUPCONFIG.XML. The following is a description example when executing an image backup. In the CONTROLLER tag, you have to write the key that is generated in BOOTKEY.DAT.

```
<?xml version="1.0" encoding="utf-8"?>
<ROOT>
 <CONTROLLER  expires = "07/30/2016 - 19:25:34" key = "590174ed-4b69388c ">
      <CONFIG   image = "UD1:\IMG\" startmod = "backup"/>
 </CONTROLLER>
</ROOT>
```

4 Plug in the USB memory device, to which STARTUPCONFIG.XML has been saved into UD1:. Then boot the controller.
5 The operation written in XML will be executed. If execution fails, error information is written in STARTUPCONFIG.LOG. Please check the cause by referring to STARTUPCONFG.LOG.
6 If the XML command was successfully executed, STARTUPCONFIG.XML is renamed to OLDSTARTUPCONFIG.XML.

13.3.5.2 Select Start Mode

You can select the start mode by using the STARTMODE tag. Change the start mode by the value set in “value”.
The parameter to be set is as follows.

Table 13.3.5.2(a) Parameter of STARTMODE Tag

Parameter	Description
value	Set the start mode. The following value can be set. CTRL：The controller is started in the controlled start mode. JITP connection is required after starting. COLD：The controller is started in the cold start mode.

Description example of controlled start

```
<?xml version="1.0" encoding="utf-8"?>
<ROOT>
 <CONTROLLER expires = "07/30/2016 - 19:25:34" key = "590174ed-4b69388c ">
    <STARTMODE value = "CTRL" />
 </CONTROLLER>
</ROOT>
```

Description example of cold start

```
<?xml version="1.0" encoding="utf-8"?>
<ROOT>
 <CONTROLLER expires = "07/30/2016 - 19:25:34" key = "590174ed-4b69388c ">
    <STARTMODE value = "COLD" />
 </CONTROLLER>
</ROOT>
```

13.3.5.3 Configure Ethernet

You can change Ethernet settings using the ETHERNET tag. Parameters to be set are as follows.

Table 13.3.5.3(a) Parameters of ETHERNET Tag

Parameter	Description
id	Select the port that you want to change Ethernet settings. You need select one from port1 to port3.
label	Set the label of the port that you want to change Ethernet settings.
ip	Set IP address of the specified port. If the value of “ip” is set to “DHCP”, automatic setting is possible by DHCP.
router	Set IP address of router.
mask	Set sub-net mask.
hostname	Set the host name

Description example of Ethernet settings

```
<?xml version="1.0" encoding="utf-8"?>
<ROOT>
 <CONTROLLER expires = "07/30/2016 - 19:25:34" key = "590174ed-4b69388c ">
    <ETHERNET id ='port1' label='CD38A' ip = "172.168.192.10" router = "172.168.1.1" mask = "255.255.240.0" hostname = "MY_ROBOT" />
 </CONTROLLER>
</ROOT>
```

Description example of Ethernet settings (DHCP)

```
<?xml version="1.0" encoding="utf-8"?>
<ROOT>
 <CONTROLLER expires = "07/30/2016 - 19:25:34" key = "590174ed-4b69388c ">
    <ETHERNET id ='port1' label='CD38A' ip = "DHCP"/>
 </CONTROLLER>
</ROOT>
```

You can change the IP address of the device that allows the JITP connection using the PENDANT tag. The parameter to be set is as follows.

Table 13.3.5.3(b) Parameter of PENDANT Tag

Parameter	Description
ip	Set IP address of the device that allows JITP connection.

Description example of PC Jog IP address setting

```
<?xml version="1.0" encoding="utf-8"?>
<ROOT>
 <CONTROLLER expires = "07/30/2016 - 19:25:34" key = "590174ed-4b69388c ">
    <PENDANT  ip = "172.168.192.200" />
 </CONTROLLER>
</ROOT>
```

13.3.5.4 Image Backup/Image Restore

You can perform an image backup or image restore by using the CONFIG tag. Parameters to be set are as follows.

Table 13.3.5.4(a) Parameters of CONFIG Tag

Parameter	Description
image	In case of an image backup: Specify the directory to save the image backup. If the image file already exists in the specified directory, image backup is not executed. In case of an image restore: Specify the directory where the file that you want to restore was saved.
startmod	Select image backup or image restore. backup：Execute image backup. restore：Execute image restore.
dcsparam	Set this parameter only when executing an image restore. You can select whether to initialize DCS parameters or not. Default value is “1”. 0：DCS parameters are initialized. 1：DCS parameters are kept.

Description example of image backup

```
<?xml version="1.0" encoding="utf-8"?>
<ROOT>
 <CONTROLLER expires = "07/30/2016 - 19:25:34" key = "590174ed-4b69388c ">
    <CONFIG image = "UD1:\IMG\" startmod = "backup"/>
 </CONTROLLER>
</ROOT>
```

Description example of image restore

```
<?xml version="1.0" encoding="utf-8"?>
<ROOT>
 <CONTROLLER expires = "07/30/2016 - 19:25:34" key = "590174ed-4b69388c ">
    <CONFIG image = "UD1:\IMG\" startmod = "restore" dcsparam = "1" />
 </CONTROLLER>
</ROOT>
```

14 OPERATION WITHOUT MODE SWITCH

This chapter describes operations when the system does not have a Mode switch.
For the R-30*i*B Compact Plus and R-30*i*B Mini Plus controllers, the Operator Panel ad mode switch are provided as optional hardware.

Contents of this chapter

14.1 OVERVIEW

> **NOTE**
> When the Robot controller does not have a Teach Pendant, the robot should be operated in AUTO mode. The mode does not need to change from AUTO and mode select functions should not be used.

When the robot system has a Teach Pendant, the mode select function has to be selected depending on the system configuration and requirements for the system.
To change the mode, either of the following functions can be used.

- TP Mode Select function
- External Mode Select function.
- Mode switch on Operator Panel (optional hardware for R-30*i*B Compact Plus and R-30*i*B Mini Plus)

To use a Teach Pendant on R-30*i*B Compact Plus or R-30*i*B Mini Plus controllers, which do not have an operator panel, you must change the mode switch function setting using the TP Mode Select function. Please refer “14.3.2 Setup” to set up the TP Mode Select function.

To use the TP Mode Select function or the External Mode Select function on R-30*i*B Plus and R-30*i*B Mate Plus, the appropriate software option is needed. And the operator panel must be the “No mode switch” type. If the option is loaded and the operator panel has the mode switch, the alarm”SYST-351 Mode select function mismatch” occurs.

14.2 SETUP

For R-30*i*B Compact Plus and R-30*i*B Mini Plus, the function to select mode can be changed by “Mode Select Func.” item in DCS screen. Mode Select menu in DCS screen is displayed when the type of controller is R-30*i*B Compact Plus or R-30*i*B Mini Plus, or the TP Mode Select option (J768) is ordered with another type of controller

```
DCS
Mode Select                              1/3
                                      Status
 Mode Select Func. :   Teach Pendant OK
 Select Mode Type:     AUTO/T1/T2    OK

 Code number for TP Mode Select


 [ TYPE ]              [CHOICE]   UNDO
```

Table 14.2 Items in DCS Mode Select menu

Item	Description
Mode Select Func.	You can choose the method to select the mode from three functions, Mode Switch, External Input, and Teach Pendant. Modes Switch: The actual Mode Switch on the optional switch box is used to select the mode. External Input: Safety input signal (SSO[6], and SSO[7]) is used to select the mode. Teach Pendant: The TP Mode Select function is used. The mode is selected from the Teach Pendant screen. This item can be changed if controller type is R-30iB Compact Plus or R-30*i*B Mini Plus. For R-30iB Compact Plus and R-30*i*B Mini Plus, “External Input” is selected when Teach Pendant is not used. Additional “Safe I/O connect option (J568)” is needed to change mode by safety input signal (SSO[6] and SSO[7]). If the TP Mode Select option (J768) is ordered with another type of controller, this item is “Teach Pendant” and the user cannot change it.
Select Mode Type	This item is only used for the TP Mode Select function. Please refer “14.3.2 Setup”.
Status	The status of the setting parameter is displayed. OK: Setting parameter and DCS parameter are the same. CHGD: Setting parameter is changed, but not applied to DCS parameter. PEND: Setting parameter is changed and applied to DCS parameter, but controller power has not been cycled.
Code number for TP Mode Select	This item is used for only TP Mode Select function. Please refer “14.3.2 Setup”.

To use “Teach Pendant” or “External input” on R-30*i*B Compact Plus and R-30*i*B Mini Plus controllers, you must disconnect the optional switch box and connect a short connecter. If “Teach Pendant” or “External input” is selected and the optional switch box is connected, the alarm”SYST-351 Mode select function mismatch” occurs.

Procedure 14-1 Change mode select function for R-30*i*B Compact Plus

Steps

1. Press the [MENU] key. Menu items are displayed.
2. Move the cursor to ‘6 SYSTEM’, and select ‘DCS’ from the fly-out menus.
3. Select ‘Mode Select’ and press ENTER key. The Mode Select screen is displayed.
4. Change Move the cursor to item “Mode Select Func.
5. Push F4[CHOICE] and choose used a modes select function.
6. Press PREV key and display DCS top screen.
7. Press F2 APPLY. Input Code number to apply new DCS parameter.
8. Re-power the controller.

> ⚠ **WARNING**
> If a robot is used with an incorrect safety setting, the safety function does not work correctly and serious personal injury could result. When the setting of the mode select function is changed, the setting must be verified and the function must be tested again.

14.3 TP MODE SELECT FUNCTION

Teach Pendant mode select function can select the operation mode, AUTO, T1 or T2, in Teach Pendant Screen.

The mode is selected on the Teach Pendant screen. Input of a passcode for this function is needed to change the mode. The Mode is changed immediately by operations on the screen.
When AUTO is select by the Teach Pendant, the servo power turns off with the “SRVO-483 Input AUTO confirmation signal”. It is not reset until the confirmation signal is changed from OFF to ON.
When the controller power up during the selected mode is AUTO, input of AUTO confirmation is needed. Any digital input can be used as the confirmation signal. The confirmation signal is defined as “Confirmation input for AUTO” on the “System Config” menu.

> ⚠ **WARNING**
> The system integrator is responsible for designing the robot system such that it complies with following requirements.
> - To use this function, a switch to be used as the AUTO confirmation signal is needed. The AUTO confirmation signals shall be connected to a switch located outside of the safeguarded space, but in the vicinity of the space.
> - Install an interlocking device, a safety plug, and so forth in the safety gate so that the robot is stopped as the safety gate is opened.
>   Operators shall lock the safety gate in the open position with a padlock when they enter the safeguarded space, depending on a risk assessment.

> ⚠ **WARNING**
> The AUTO Confirmation Switch shall only be pushed after it is confirmed that nobody is in the safeguarded space.

The mode is selected when the controller starts up as follows.
- When the controllers power down in T1, the mode is kept T1 at start-up.
- When the controllers power down in T2, the mode is set to T1 at start-up.
- When the controllers power down in AUTO, the mode is kept as AUTO at start-up. Input of the confirmation signal is needed to reset the SRVO alarm and to start moving the robots.

When the Teach Pendant is disconnected, the controller retains the selected mode. Connecting to the Teach Pendant is needed to change the mode.

The safe I/O SSI[7](AUTO), SSI[8](T1) and SSI[9](T2) is used to choose the current selected mode when the Teach Pendant mode select function is used. These safe I/O are safety signals rated as Cat.4, PL e.

To use this option on R-30*i*B Plus and R-30*i*B Mate Plus, the operator panel must be set to “No mode switch” type. If the option is loaded and the operator panel has the mode switch, the alarm ”SYST-351 Mode select function mismatch” or ”SYST-301 External mode select is not available” occurs.

To use this option on R-30*i*B Compact Plus or R-30*i*B Mini Plus, disconnect the optional mode switch and connect a special short connecter. If the option is enabled and the hardware is not matched, the alarm the "SYST-351 Mode select function mismatch" occurs.
The mode cannot be changed from the PC or tablet device screen (by the Remote *i*Pendant function). This pop-up screen does not appear even if TP Enable switches on the PC screen is changed.

14.3.1 Operation

The pop-up screen to change the mode appears when the TP Enable switch is changed.
The pop-up screen is displayed in the following cases.
- The selected Mode is AUTO and the TP Enable switch is changed to ON.
- The selected Mode is T1 or T2 and the TP Enable switch is changed to OFF.

The mode is changed when another mode button of other mode is pressed, or if the cursor is moved and the ENTER key is pressed.
When CANCEL is selected or the PREV key is pushed, the pop-up screen disappears and the selected mode is not changed

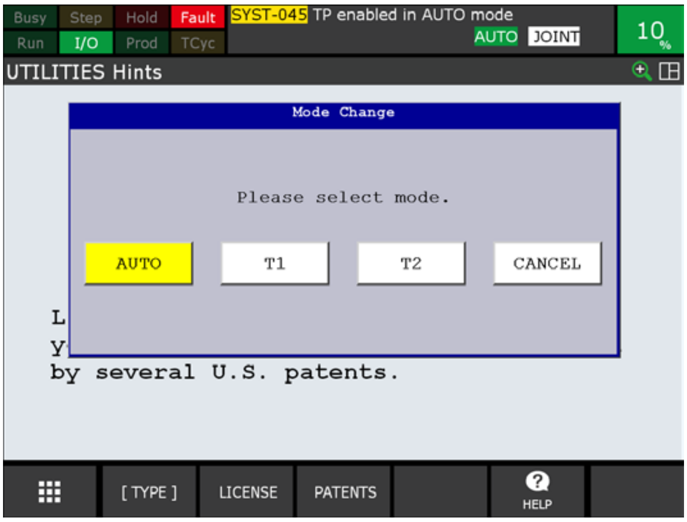


Fig. 14.3.1 (a) Mode select pop-up screen

A master code number input line is displayed after another mode is selected. To change the mode, input the code number. The default code number is "1111". The code number can be changed in DCS Mode Select screen.
If the code number is correct, the mode will change to the selected mode. If the code number is not matched, the current mode is kept and "PWD-098 Invalid Password" is displayed in the alarm space.

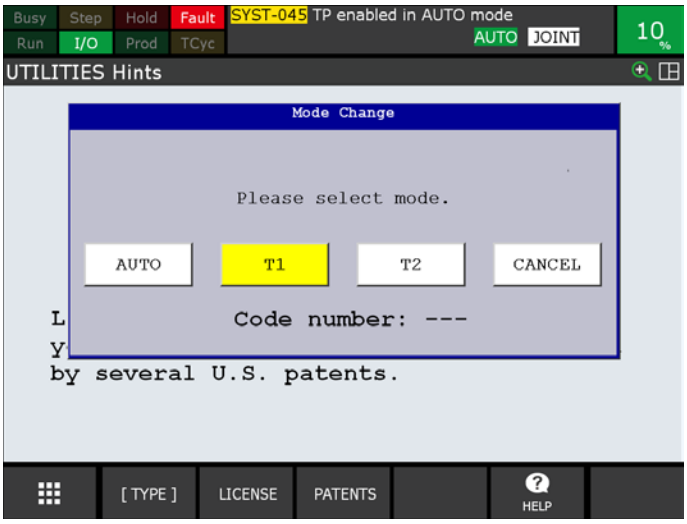


Fig. 14.3.1 (b) Mode select pop-up screen to input code number

When AUTO is selected, “SRVO-483 Input AUTO confirmation signal” occurs. Push the switch that is set as the AUTO confirmation signal to reset this alarm.

> ⚠ **WARNING**
> Confirm that no one is in the safeguarded space before you press the AUTO Confirmation Switch.

14.3.2 Setup

If controller type is R-30*i*B Compact Plus or R-30*i*B Mini Plus, or the TP Mode Select option (J768) is ordered, the “DCS Mode Select screen” will be displayed in the DCS screen.

```
DCS
Mode Select                                  1/3
                                          Status
 Mode Select Func. :   Teach Pendant  OK
 Select Mode Type:     AUTO/T1/T2     OK

 Code number for TP Mode Select


 [ TYPE ]                [CHOICE]    UNDO
```

Table 14.3.2 Items in DCS Mode Select menu

Item	Description
Mode Select Func.	You can choose the method to select the mode from three functions; Mode Switch, External Input, and Teach Pendant. Modes Switch: An actual Mode Switch on an optional switch box is used to select the mode. External Input: Safety input signal (SSO[6], and SSO[7]) is used to select the mode. It works the same as the “External Mode Select function (J569). Teach Pendant: This function (TP mode select function) is used to select the mode. This item can be changed if the controller type is R-30*i*B Compact Plus or R-30*i*B Mini Plus. The default setting is “External Input” for the R-30*i*B Compact Plus and R-30*i*B Mini Plus. If the TP Mode Select option (J768) is ordered with another type of controller, this item is “Teach Pendant” and the user cannot change it.
Select Mode Type	You can select whether the T2 button is displayed in the pop-up screen. AUTO/T1: AUTO and T1 buttons are displayed in the pop-up screen. AUTO/T1/T2: AUTO, T1 and T2 buttons are displayed in the pop-up screen. (Default)

Status	The status of the User model is displayed. OK: Setting parameter and DCS parameter are the same. CHGD: Setting parameter is changed, but not applied to the DCS parameter. PEND: Setting parameter is changed and applied to the DCS parameter, but controller power has not been cycled.
Code number for TP Mode Select	A password to protect the mode change is changed in the DCS Mode Select screen. Only a four numbers code can be used for the code to protect the mode change just like other DCS code numbers (For example, 9999, 1234 or etc.). The default numbers is “1111”. To change the code number, you will need to input the DCS master code number.

> ⚠**WARNING**
> If a robot is used with an incorrect safety setting, the safety function will not work correctly and serious personal injury could result. When the setting of the mode select function is changed, the setting must be verified and the function must be tested again.

Procedure 14-2 Change of the code number of TP Mode Select

Steps

1. Display the DCS Mode Select screen.
2. Move the cursor to “Code number for TP Mode Select“ and push the [ENTER] key
3. When the message "Code number (Master):" is displayed, enter the DCS master code number
4. When the message " Enter new code number: " is displayed, enter the new code number for the TP Mode Select.
5. When the message "Verify new code number:" is displayed, enter the new code number for the TP Mode Select again for verification.

The code number for TP Mode Select is saved as part of the robot backup in SYSPASS.SV. When the code numbers are changed by loading the SYSPASS.SV file, the alarm "SYST-219 Need to apply DCS param" occurs. To clear the alarm, the “Apply to DCS parameter” operation must be performed. In this operation, the previous master code number must be entered as follows.

- The message "Previous code number (master):" is displayed. Please enter the master code number that was used before SYSPASS.SV was loaded.
- When the message "Code number (Master)" is displayed, please enter the master code number that is loaded from the SYSPASS.SV.

Confirmation input for AUTO

Confirmation input for AUTO is defined in the System Configuration screen. Index of the DI signal is set in the screen. When the selected mode is changed to AUTO, a SRVO alarm occurs. To reset the alarm, this switch has to be also changed from OFF to ON. If this is not done setting, the SRVO alarm cannot be reset.

```
System/Config
                                          1/60
1 Use HOT START:                     TRUE
         ...
56  No motion PR operate mode:       FALSE
57  Use No-Step-In TP Programs:      FALSE
58  Export diagnosis data:         <*DETAIL*>
59  Check Sim. I/O when FWD/BWD:     FALSE
60  Confirmation for AUTO:          DI[    1]

[ TYPE ]
```

> ⚠ **WARNING**
> The AUTO confirmation signals shall be connected to a switch located outside the safeguarded space, but in the vicinity of the space.

14.4 EXTERNAL MODE SELECT FUNCTION

Safe I/O SSO[6] and SSO[7] are used to select the AUTO/T1/T2 operation mode in place of a mode switch on operator panel.
To use this option, the operator panel must be "No mode switch" type.
For the R-30*i*B Compact Plus and R-30*i*B Mini Plus, Safe I/O connect option (J568) is needed to change mode by safe I/O.

Table 14.4 Safety I/O for Mode Select

Index	Name	Description
SSO[6]	C_T1	SSO[6] and SSO[7] are used to select AUTO/T1/T2 operation mode in place of mode switch on operator panel. To use SSO[6] and SSO[7], "External mode select" option (A05B-2600-J569) is necessary and the operator panel must be "No mode switch" type. If the option is loaded to a robot controller that has the operator panel mode switch, the alarm "SYST-301 External mode select is not available" occurs. AUTO/T1/T2 mode is selected according to the status of SSO[6] and SSO[7] as follows. SSO[6:C_T1] SSO[7:C_T2] Mode OFF OFF AUTO ON OFF T1 ON ON T2 OFF ON Invalid (SYST-037, SRVO-411) Note: When "External mode select" option is used, a risk assessment for the whole robot system is necessary, including the means to select the operational mode.
SSO[7]	C_T2	
SSI[7]	AUTO	ON: AUTO mode, OFF: Not AUTO mode
SSI[8]	T1	ON: T1 Mode, OFF: Not T1 Mode
SSI[9]	T2	ON: T2 Mode, OFF: Not T2 Mode

> ⚠**WARNING**
> When the "External mode select" option is used (AUTO/T1/T2 operational mode is selected by SSO[6] and SSO[7]), a risk assessment for the whole robot system is necessary, including the means to select the operational mode.
> If a robot is used with an incorrect operational mode, the safety function does not work correctly and serious personal injury may result.
> The means to select the operational mode must fulfil the following requirement.
>
> Operational modes shall be selectable with a mode selector which can be locked in each position (e.g. a key operated switch which can be inserted and extracted in each position). Each position of the selector shall be clearly identifiable and shall exclusively allow one control or operating mode.

14.4.1 Setup

To use External Mode Select function on R-30*i*B Compact Plus and R-30*i*B Mini Plus, "Mode Select Func." in DCS menu is change to "External Input" and the setting is applied as DCS parameter.
For R-30*i*B Plus, option (J569) is needed to use External Mode Select function.

14.5 OPTIONAL MODE SWITCH

A Mode switch is provided as an optional switch box on the R-30*i*B Compact Plus and the R-30*i*B Mini Plus. The mode switch on the optional switch box can change the mode to one of three modes (AUTO, T1 and T2). Other types of controllers have mode switch is implemented on an operator panel by the default.

14.5.1 Setup

To use the optional mode switch on the R-30*i*B Compact Plus and R-30*i*B Mini Plus, “Mode Select Func.” in the DCS menu nees to change to “Mode Switch” and the setting is applied to the DCS parameter.

15 OPERATIONS TABLET TEACH PENDANT

This chapter describes operations when the system had a Tablet Teach Pendant.
Software option (S527) is needed to use Tablet Teach Pendant.

Contents of this chapter

15.1 OVERVIEW

This function can use a tablet as Teach Pendant. Software option (S527) is needed to use Tablet Teach Pendant. Also, Tablet Base (A05B-2256-K301/A05B-2256-K302) is needed to connect Tablet and the controller.

Tablet must have USB Type-C port. If tablet does not have USB Type-C port, tablet cannot connect to Tablet Base.

Supported OS of Tablet device are follows.
- Android
- iOS

If Android tablet is used, dedicated application can be used.
This function works on a web browser if dedicated application is not used. The following browser application is supported.
- Google Chrome

15.2 SETUP

15.2.1 TCP/IP setting

To connect to the controller, you must set the TCP/IP setting on your tablet.

You will need to set the IP address of your tablet device to access the controller.

IP address: 1.1.0.12
Subnet Mask: 255.255.255.0

15.2.2 Tablet TP APP

Tablet TP APP can be downloaded from the controller if using Android tablet. After connecting the controller, you can access the following URL when using web browser and install Tablet Teach Pendant APP.

URL: http://1.1.0.10/frh/irprog/android/tablettp.apk

15.3 OPERATIONS

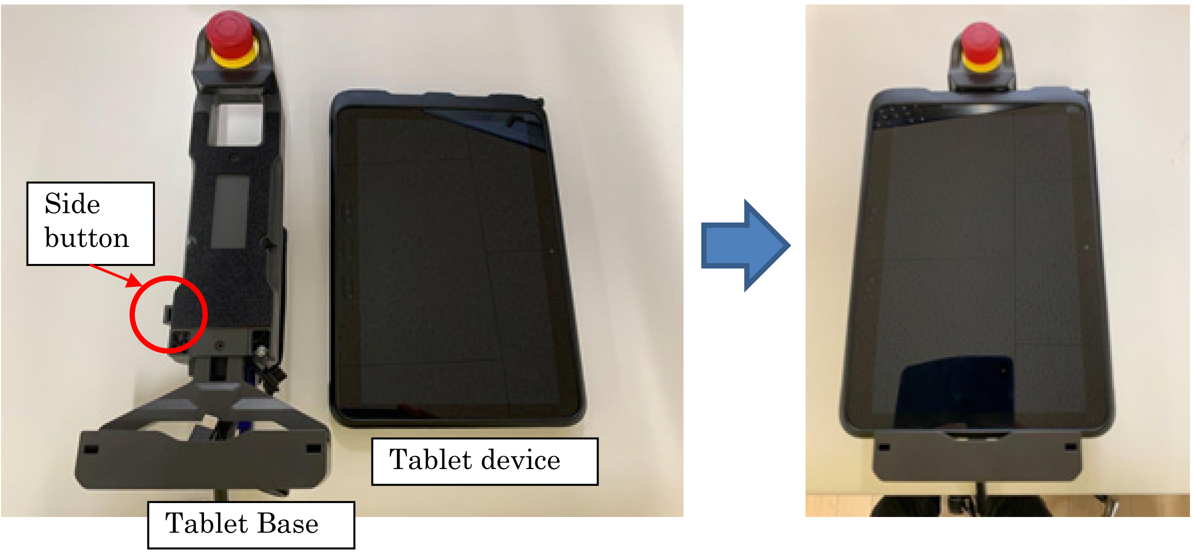
15.3.1 Connect to Controller

Connect tablet and the controller via Tablet Base.

Procedure 15-1 Connection to Tablet Teach Pendant and login to controller

Steps

1 Connect TP cable to Tablet Base.
2 Combine tablet device and Tablet Base. Press the side button and stretch Tablet Base. Fit tablet device into Tablet Base



3 Turn on the controller
4 Start Tablet Teach Pendant APP. You can login the controller if the controller completely starts.

If you don't have Tablet TP APP, you can connect the controller from Google Chrome APP. Access to the robot homepage (http://1.1.0.10/) and select iRProgrammer menu. You can login the controller.

15.3.2 Emergency Stop

Tablet Teach Pendant has an emergency stop button (TP E-stop) on Tablet Base. If Tablet Teach Pendant is connected to the controller, the following alarm is occurred and the robot motion is stopped when press each Estop button.

Table 15.3.2(a) Alarm of Emergency Stop

Controller	button\signal	Alarm
R-30iB Plus	TP E-stop	SRVO-601 TP/OP E-stop
	OP E-stop	SRVO-601 TP/OP E-stop
	EX E-stop	SRVO-007 External emergency stops
R-30iB Mate Plus	TP E-stop	SRVO-602 TP/External E-stop
	OP E-stop	SRVO-001 Operator panel E-stop
	EX E-stop	SRVO-602 TP/EX E-stop
R-30iB Compact Plus	TP E-stop	SRVO-002 Teach pendant E-stop
	OP E-stop	SRVO-001 Operator panel E-stop
	EX E-stop	SRVO-007 External emergency stops
R-30iB Mini Plus	TP E-stop	SRVO-002 Teach pendant E-stop
	OP E-stop	SRVO-001 Operator panel E-stop
	EX E-stop	SRVO-007 External emergency stops

Please refer to “FANUC ROBOT SAFETY HANDBOOK 4.3 STOP TYPE OF ROBOT” for details of stop type of the robot motion.

15.3.3 Enabling Device (Deadman switch)

Tablet Teach Pendant has an enabling device (Deadman switch) on Tablet Base. When Tablet TP is connected, the following alarm is occurred.

Table 15.3.3(a) Alarm of Enabling device

Controller	Enabling device	Alarm
R-30iB Plus	Enabling device (Deadman switch)	SRVO-603 Deadman switch/NTED released
	NTED	SRVO-603 Deadman switch/NTED released
R-30iB Mate Plus	Enabling device (Deadman switch)	SRVO-003 Deadman switch released
R-30iB Compact (Mini) Plus	Enabling device (Deadman switch)	SRVO-003 Deadman switch released
R-30iB Mini Plus	Enabling device (Deadman switch)	SRVO-003 Deadman switch released

Please refer to “FANUC ROBOT SAFETY HANDBOOK 4.3 STOP TYPE OF ROBOT” for details of stop type of the robot motion.
When Tablet Teach Pendant is enabled, robot motion is allowed only while enabling device (deadman switch) is gripped.

15.3.4 Screen

Tablet Teach Pendant has the following elements that are different from *i*Pendant.

- Status bar
- Menu
- Key Sheet

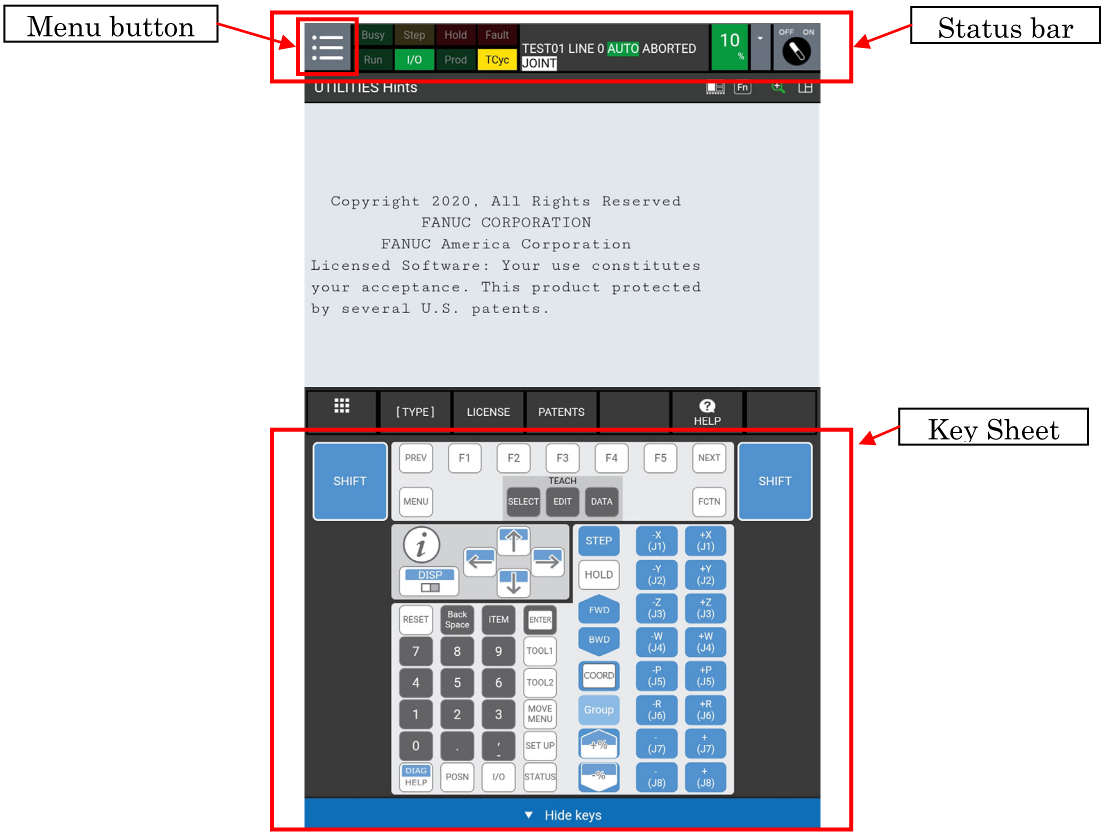
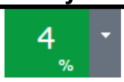


Fig 15.3.4(a) Tablet Teach Pendant Screen

15.3.4.1 Status bar

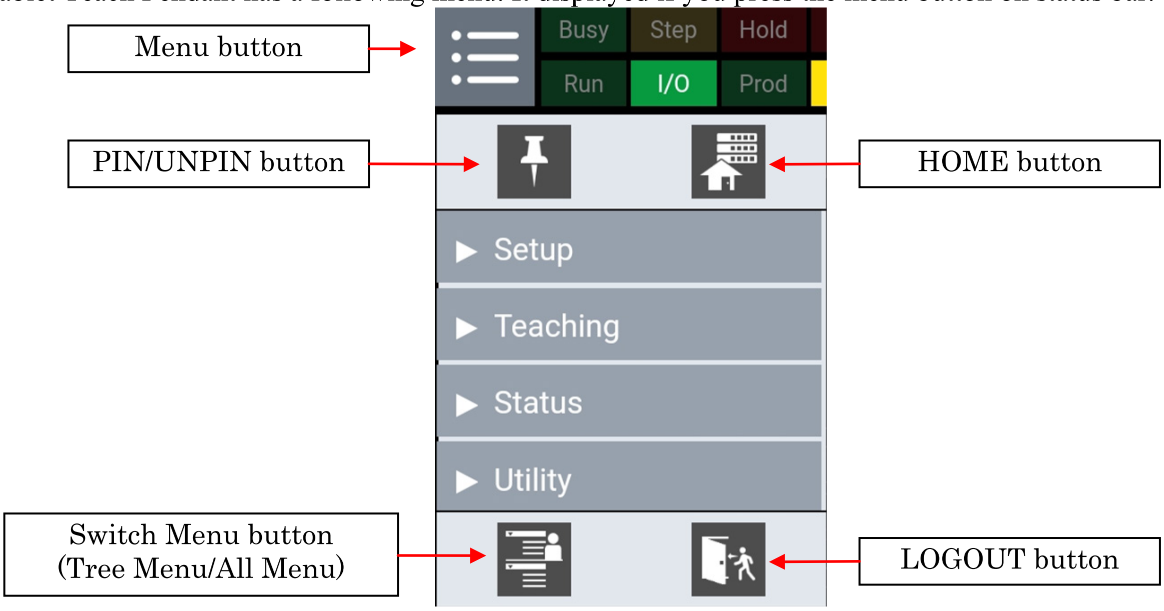
Status bar has the following keys.

Table 15.3.4.1(a) Keys in Status bar

Keys	Description
 	Slider to change override value. The slider is displayed when an arrow button is clicked.
	The TP enable switch allows you to move the robot using Tablet Teach Pendant. When the key is ON, the robot can be moved by jogging and program execution.
	The menu key to display tree menu.

15.3.4.2 Menu

Tablet Teach Pendant has a following menu. It displayed if you press the menu button on status bar.



PIN/UNPIN button

If PIN button is pressed, fixed menu is displayed on left side.
Menu button is not disappeared from status bar.
If UNPIN button is pressed, pull down menu is displayed.

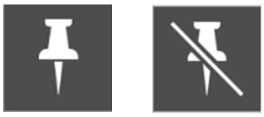


Fig 15.3.4.2 (a) PIN button (left) / UNPIN button (right)

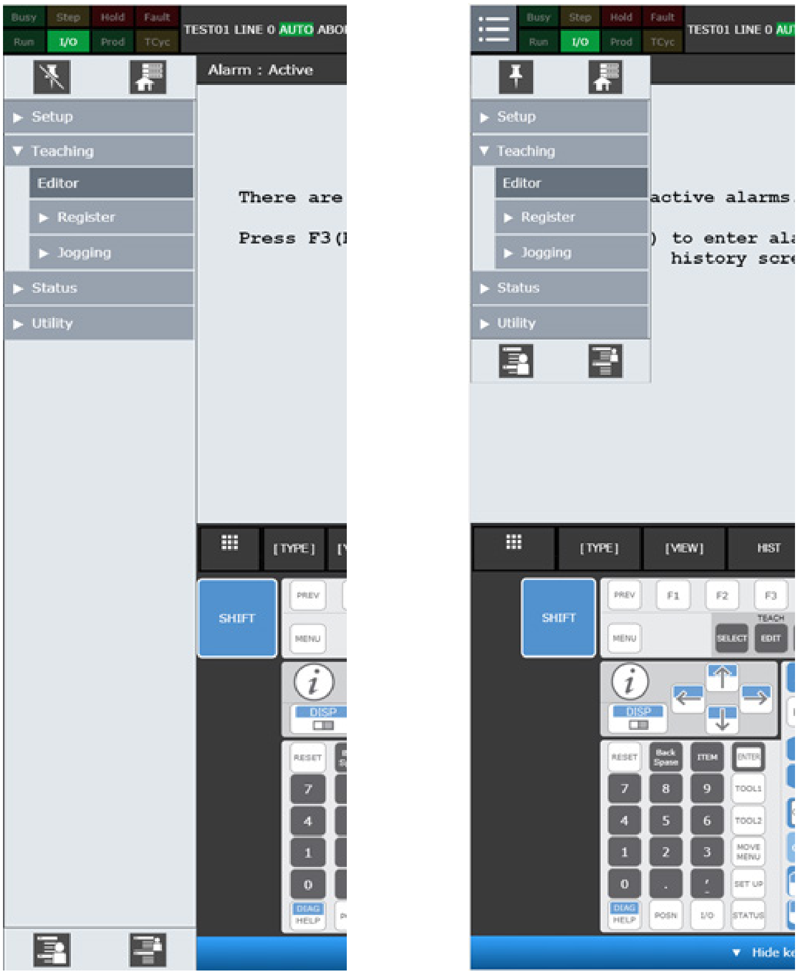


Fig 15.3.4.2 (b) Fixed menu (left) / Pull down menu (right)

HOME button

If HOME button is pressed, iHMI HOME screen is displayed.

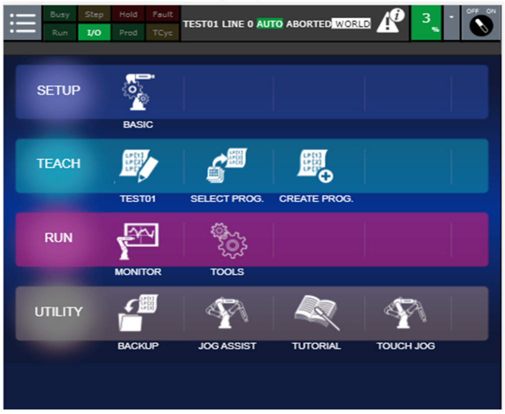


Fig 15.3.4.2 (c) *i*HMI Home

Switch Menu button

If Switch Menu button is pressed, menu is switched to “Tree Menu” or “All Menu”.
“All Menu” is conventional TP menu.
“Tree Menu” has the following items.

Table 15.3.4.2(a) Tree Menu

Section	Menu1	Menu2	Remark
Setup	Initial Setup Guide		Initial setup
	Frame	Utool	Tool Frame setting
		Utool Guide	Tool Frame setup Guide
		Uframe	User Frame setting
	Motion	Payload	Payload setting
		Reference Position	Reference Position setting
		Axis Limit	Axis Limit setting
	Network	Host Comm	Network setting
	Comments	String Register	String Register comment setting
		User Alarm	User alarm comment setting
	Others	Program Setup	Setting for program execution
		General Setup	General setting (language etc.)
		DCS	DCS setting
Teaching	Editor		Programming Editor
	Register	Numeric Register	Show and set numeric register
		Position Register	Show and set or record position register
	Jogging	Touch Jog	Jogging by using Touch Jog screen
Status	Current Position		Display current position
	Alarm Status		Display error status
	I/O		Show and set I/O value
Utility	File		Display files in devices
	File Backup Guide		Backup Guide
	Image Restore Guide		Image Backup Restore Guide
	Cycle Power		Select start mode for next startup

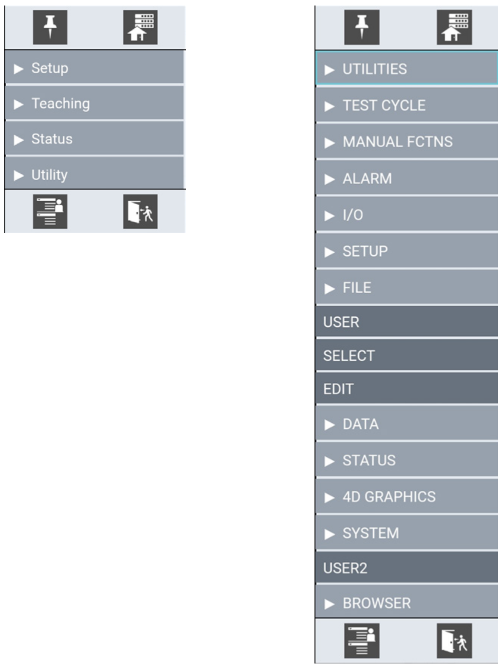


Fig 15.3.4.2 (d) Tree menu (left) / All Menu (right)

Logout button

If you press Logout button, Tablet Teach Pendant is disconnect from the controller.

15.3.4.3 Key sheet

Tablet Teach Pendant has Key sheet that imitates *i*Pendant. Key sheet can be displayed if you press show keys button. Please refer to “2.3.1 Teach Pendant” for details of each keys.

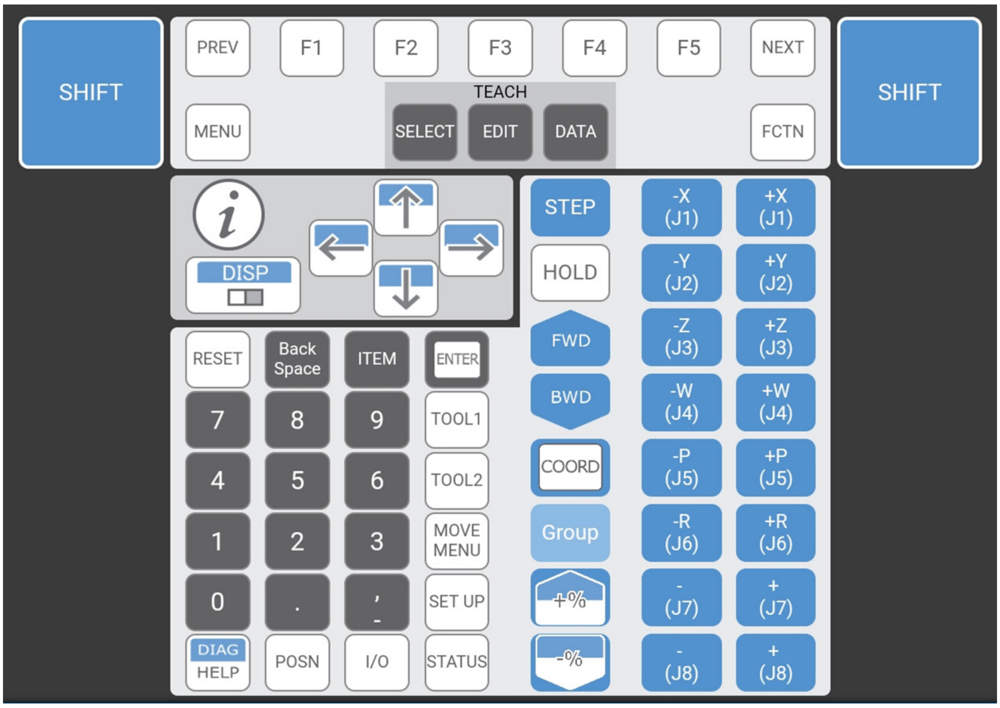


Fig 15.3.4.3(a) Key Sheet

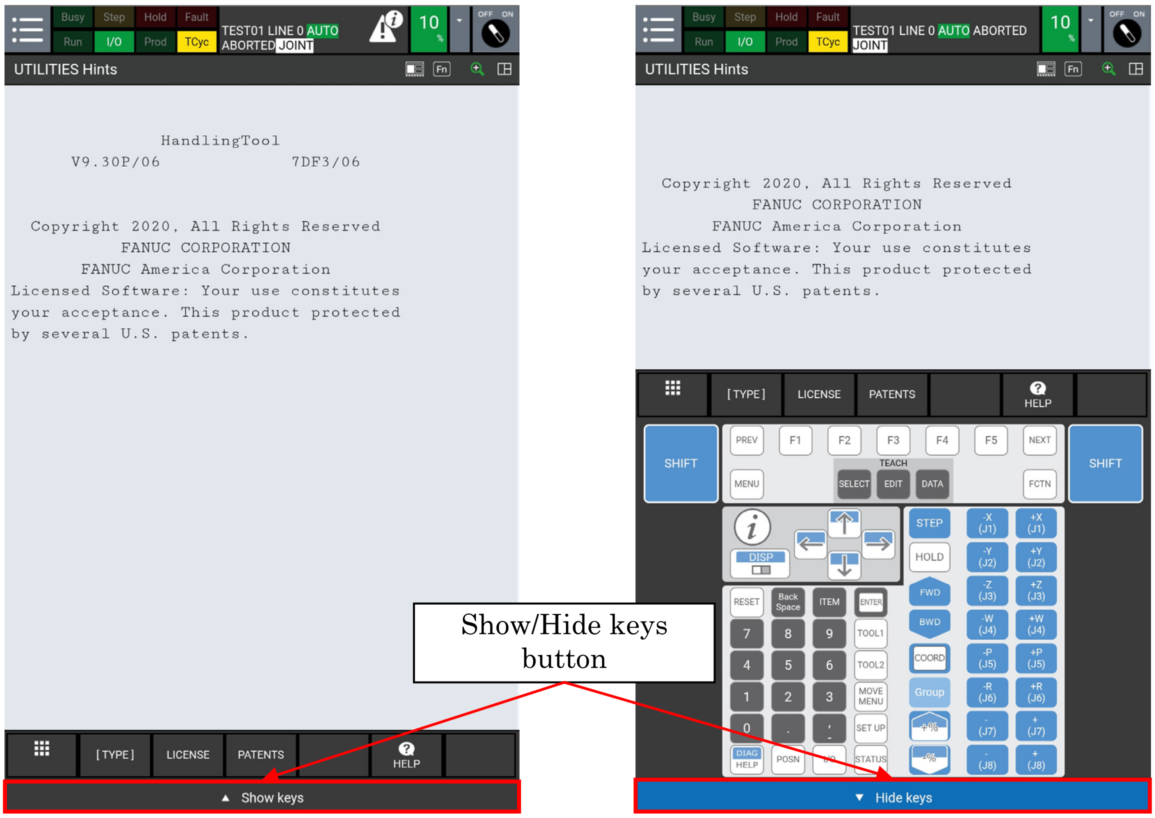


Fig 15.3.4.3(b) Show/ Hide keys

Some keys are used with SHIFT key. You can press SHIFT key in the following two ways.

- Multi-tap SHIFT key and another key that need SHIFT.
- Double-tap the key that need SHIFT.

Function key area on TP screen corresponds to F1 to F5 key on Key Sheet.
If you want to press SHIFT and Function key area, the same operation can be performed by pressing SHIFT and F1 to F5 key.

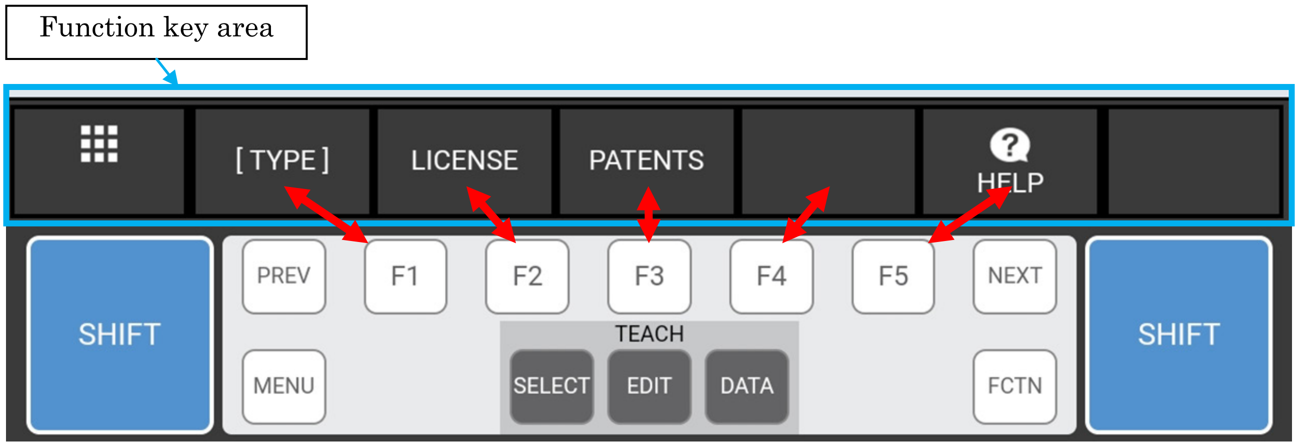


Fig 15.3.4.3(c) Function key

15.3.5 Jog Operation and Program Execution

Tablet Teach Pendant has enable switch on Tablet Base. If you want to jog the robot or execute the program with Key sheet, you should press the enable switch while jogging the robot or running the program. Enable switch must be pressed even if AUTO mode is selected.

15.3.5.1 Jog operation

If you want to jog the robot, you have to press SHIFT and Jog key. You can press SHIFT and JOG key in the following two ways.
- Tap Jog Key while pressing SHIFT key.
- Double-tap Jog Key.

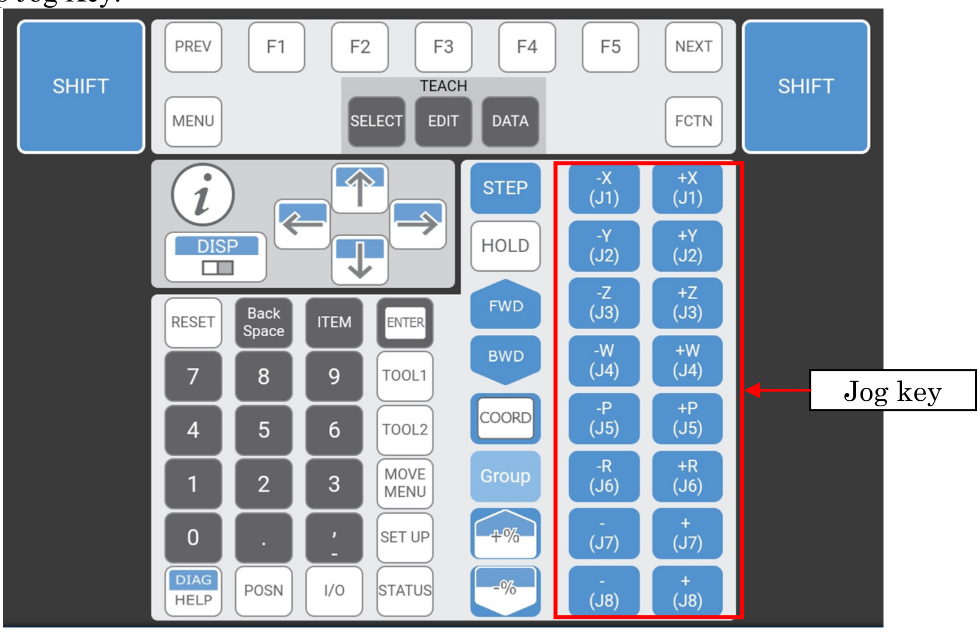


Fig 15.3.5.1(a) Jog key

Please refer to “5.2.3 Moving the Robot by Jog Feed” for details of Jog operation.

> ⚠ **WARNING**
> Jog operation must be executed with the robot in sight.

15.3.5.2 Program execution

If you want to execute the program, you have to press SHIFT and FWD/BWD key. You can press SHIFT and FWD/BWD key in the following two ways.

- Tap FWD or BWD Key while pressing SHIFT key.
- Double-tap FWD or BWD Key.

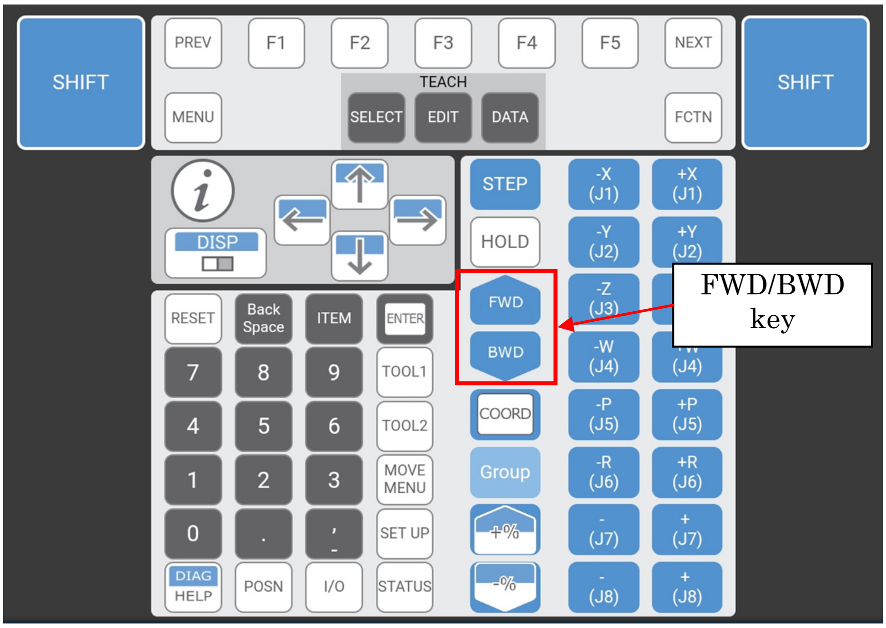


Fig 15.3.5.2(a) FWD/BWD key

> ⚠ **WARNING**
> Verify the selected program and line number is correct before running a program.

> ⚠ **WARNING**
> When you start running programs using Tablet Teach Pendant in AUTO mode, you must strictly adhere to the following precautions.
> - Running programs must be executed from outside of the safeguarded space.
> - You must check that no one is inside in the safety fence.

Please refer to “6.2 EXECUTING A PROGRAM” for details.

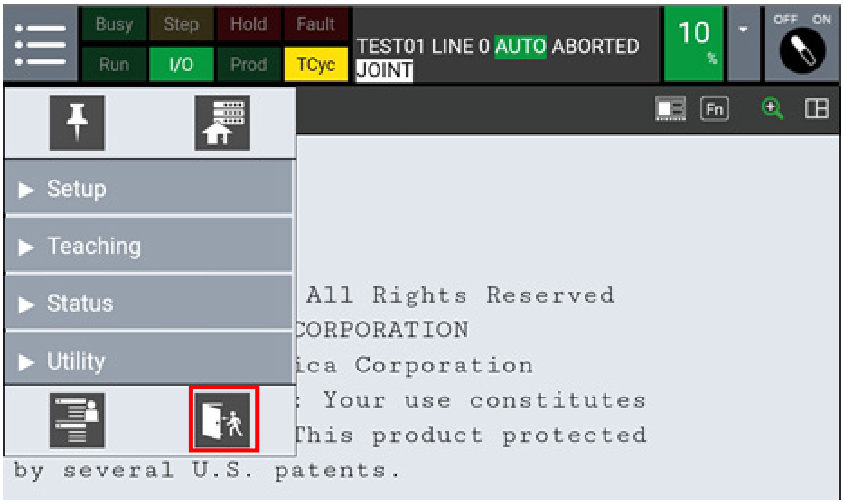
15.3.6 Disconnect from Controller

Disconnect Tablet Teach Pendant from the controller in the following steps.

Procedure 15-2 Disconnect Tablet Teach Pendant from the controller

Steps

1. Press menu button on status bar.
2. Press Logout button. If you use Tablet TP APP, it will close.



15.4 PLUGIN FUNCTION

15.4.1 Overview of Plugin Function

The Plugin function provides the way for the user to load the Plugin software, which is provided by a peripheral equipment supplier, to a FANUC robot. Together with the standardized mechanical and electrical interface, this feature makes the connection and installation of peripheral equipment simple and easy.

The figure 15.4.1 (a) shows the overview of the plugin software. This uses a gripper as an example, but the plugin may be provided for any application, such as sensor system, vision system, and so forth.

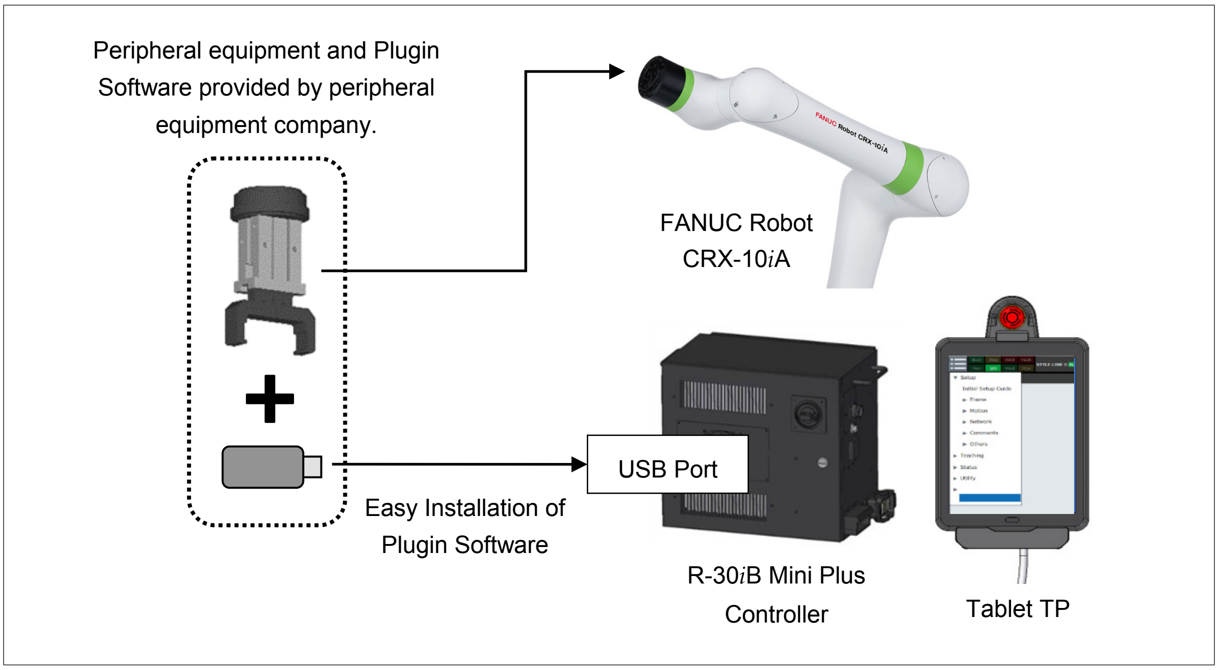


Fig 15.4.1 (a) Overview of Plugin Software Installation

The following two screens are what the user would see. Installation of the plugin software, and its management can be done in these screens.

Install screen

This screen enables the user to install the Plugin software using a USB memory.

Plugin List screen

This screen shows a list of the installed plugins and detail profile information. This screen also let you uninstall the installed plugin.

> ⚠ **CAUTION**
> When connect the peripheral device compatible to Plugin, please ensure to complete the installation of plugin software before connecting the device to EE connector. If you connect the device before installation, the device might be broken.

> ⚠ **CAUTION**
> When insert or remove the peripheral device connected to the EE connector, please ensure to turn off the robot controller.

15.4.2 Install Screen of Plugin

A plugin software that has been developed by a peripheral equipment provider can be easily installed using a USB memory. Installation of plugin software is done in the “Install” screen. The screen shows the detail information of the plugin software to be installed, such as the name of the provider of the plugin, the name of the equipment supported, software version number, etc. By installing a plugin software, new screens specifically for the equipment and new program instructions to control the equipment may be added, and also the system may be set up for the equipment, such as tool frame, payload setting, etc.

The following shows the procedure to install a plugin.

Procedure Installing a Plugin Software

Procedure

1 Tap the Menu button on the Tablet TP, and you will see the pull-down menu similar to the following.

Fig 15.4.2 (a) Pull-down menu displayed when the menu button is tapped (English / Japanese)

2 Tap the “Install” item under “Plugins”, and you will see the “Install” screen as shown in the Figure 15.4.2 (b). The screen shows the information that is stored in the USB memory, such as the name of the plugin install package file, and the detail information of the plugin software.

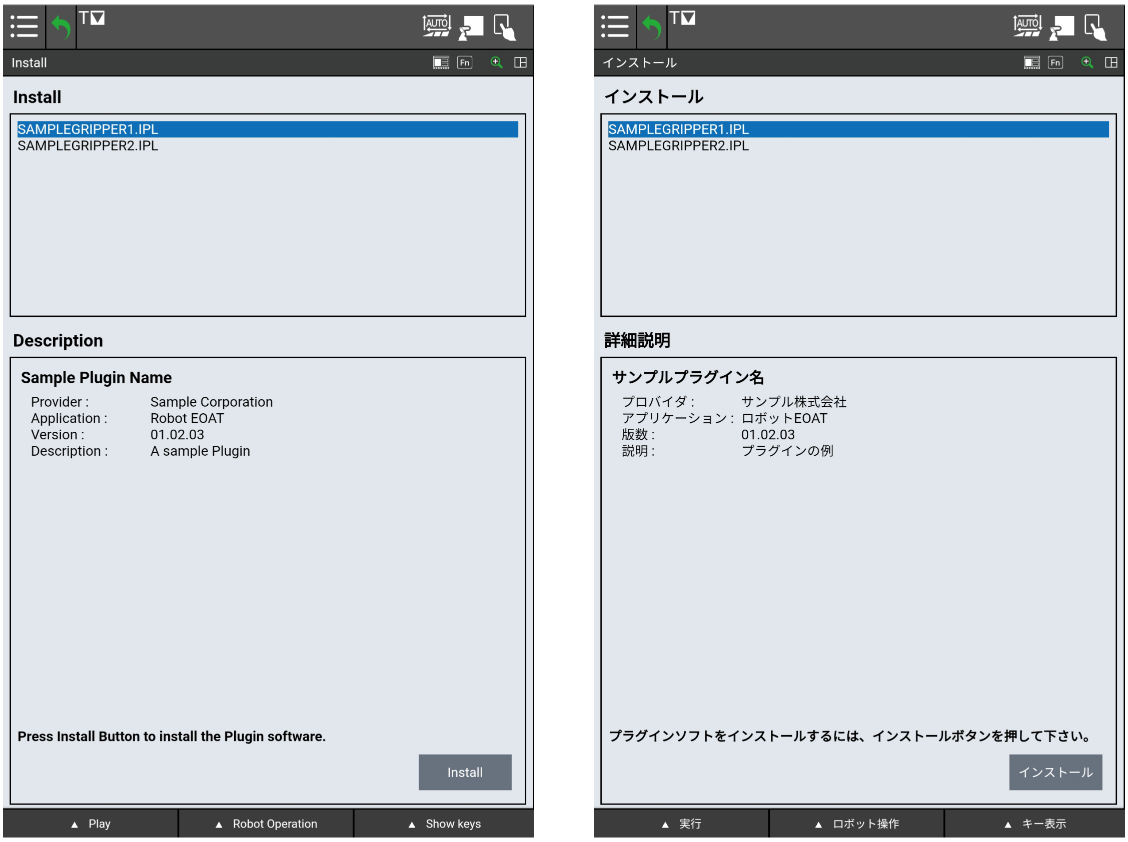


Fig 15.4.2 (b) Install Screen (English / Japanese)

3 Tap the “Install” button the bottom to start the installation processing.
4 The following screen will be displayed when the installation processing is done.

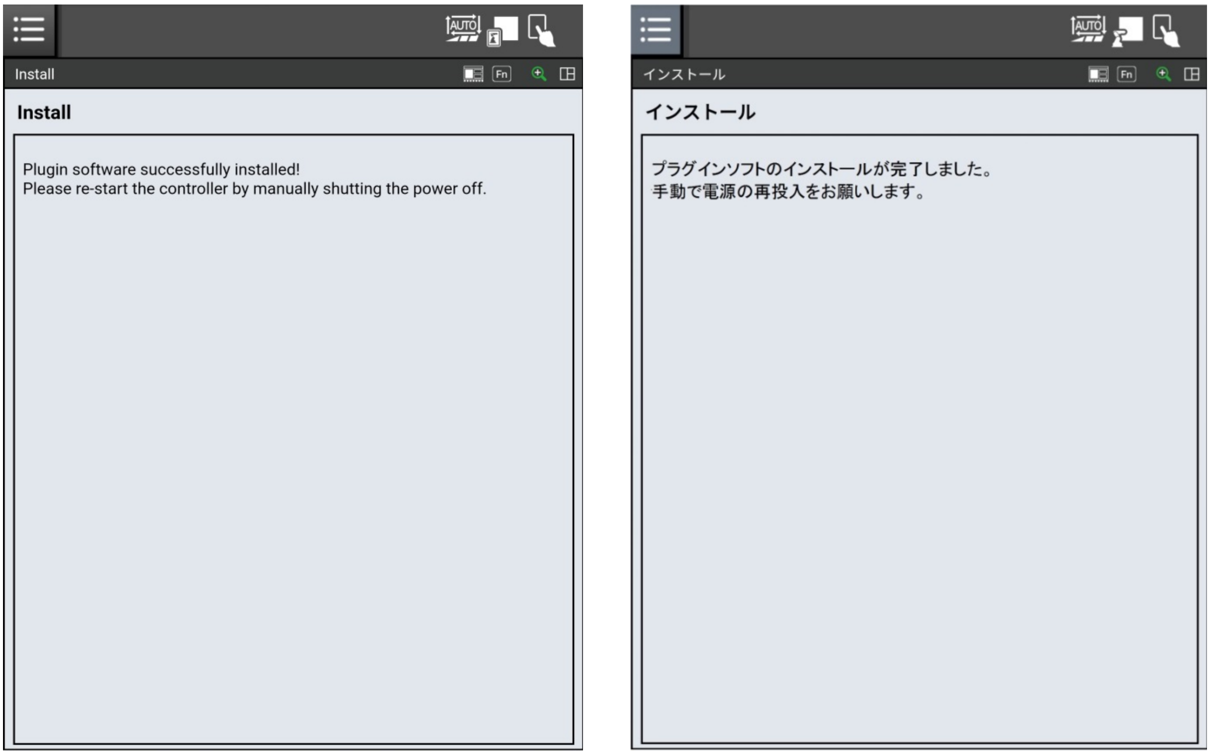


Fig 15.4.2 (c) Screen showing Installation completion (English / Japanese)

5 Please re-start the controller by manually shutting down the controller power.

15.4.3 Plugin List Screen

The Plugin List Screen shows a list of installed Plugin software with the detail information of each plugin. You may also uninstall (delete) a plugin software on this screen.

1. A list of installed plugin software
2. Detail information of a selected plugin software
3. Uninstall operation of an installed plugin software

Procedure Displaying a List of Installed Plugin Software

Procedure

1 Press the Menu button on the Tablet TP, and you will see the pull-down menu similar to the following.

Fig 15.4.3 (a) Pull-down menu displayed when the menu button is tapped (English / Japanese)

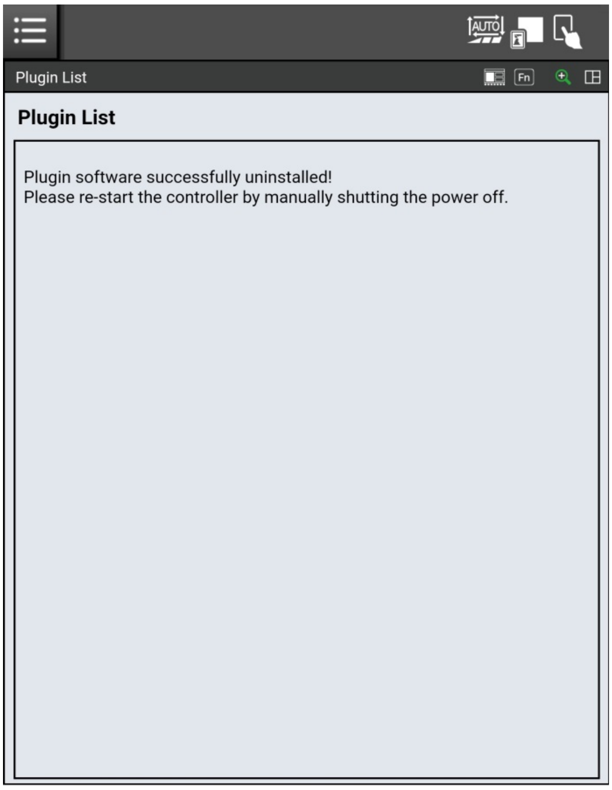
2 Tap the “Plugin List” item under “Plugins”, and you will see the “Plugin List” screen as shown in the Figure 15.4.3 (b).

Fig 15.4.3 (b) Plugin List screen (English / Japanese)

3 Tap an item in the list to select a plugin software, and you will see the detail information of the selected plugin in the lower half of the screen.

4 You can uninstall the selected plugin by tapping the “Uninstall” button on the bottom right of the screen.

5 The following screen will be displayed when the uninstallation processing is done. Please re-start the controller by manually shutting down the controller power.



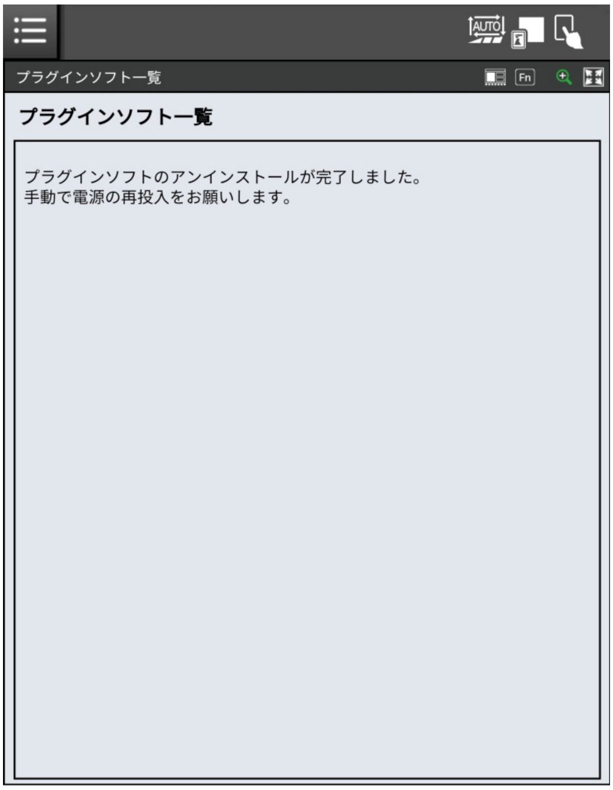


Fig 15.4.3 (c) Completion of Uninstall Operation (English / Japanese)

15.4.4 Limitations of Plugin Function

The following limitations apply to Plugin Function.

- This feature is only available for 7DF5 series (V9.40P) software.
- Tablet TP is required for this feature.
- This feature is only available on FANUC Robot CRX-10*i*A, CRX-10*i*A/L.

15.5 LIMITATIONS

The following limitations apply to Tablet Teach Pendant.

- Supported OS are Android and iOS.
- Tablet TP APP for Android is supported version 9.0.
- Tablet device must have USB Type-C port.
- Screen size of device is recommended more than 9 inch to use Tablet Teach Pendant.
- Supported languages are English, Japanese and Chinese. The screen is displayed at English when other language is selected.
- Supported languages for Key Sheet are English and Japanese. Key Sheet is displayed at English when other language is selected.
- Multi-tap operation with touch pen is not supported.

APPENDIX

A SCREEN/PROGRAM INSTRUCTIONS

This appendix summarizes the items necessary for using this model. It may also be used as an index.

Contents of this appendix

A.1 LIST OF MENUS

Item	Function
1 ABORT (ALL)	Program abort
2 Disable FWD/BWD	Disabling FWD/BWD from teach pendant
3 CHANGE GROUP	Change Motion Group[*1]
4 TOGGLE SUB GROUP	Sub group toggle[*2]
5 TOGGLE WRIST JOG	Wrist jog toggle
6	
7 RELEASE WAIT	Release Wait
8	
9	
0 — NEXT —	

Fig. A.1 (a) Function menu (Page 1)

Item	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 CYCLE POWER	Power off/on
9 ENABLE HMI MENUS	Enable the HMI menu

0 — NEXT —

Fig. A.1 (b) Function menu (Page 2)

Item	Function
1 REFRES PANE	Redisplay the screen
2	
3	
4	
5	
6	
7 Diagnostic log	Save diagnostic log
8 Del Diag Log	Delete the saved diagnostic log
9	
0 — NEXT —	

Fig. A.1 (c) Function menu (Page 3)

Screen menu	Screen change menu	Screen
1 UTILITIES	Hints	Hint F5→ Help
	*i*Rcalibration	*i*Rcalibration screen
	Prog Adjust	Program Adjust Schedule List F2→ Program Adjust Schedule Detail
	Program shif	Program Shift
	Mirror Image shift	Mirror Image Shift
	Tool Offset	Tool Offset
	Frame Offset	UFrame Offset
	Angle entry shift	Angle Entry Shift
	Group Exchg	Group Mask Exchange
2 TEST CYCLE	Test Cycle	Test cycle
3 MANUAL FCNTS	Macros	Manual operation
4 ALARM	Alarm Log	Alarm occurrence ↔ Alarm history F5→ Alarm detail
	Motion Log	Motion alarm F5→ Alarm detail
	System Log	System alarm F5→ Alarm detail

	Appl Log	Application alarm F5→ Alarm detail
	Password Log	Password log F5→ Alarm detail
	Comm Log	Communication alarm F5→ Alarm detail
5 I / O	Cell Intface	I/O cell input/output screen F2→ Allocation
	Custom	Custom I/O F3→ Allocation
	Digital	Digital I/O F2→ Digital configuration F4→ Digital detail
	Analog	Analog I/O F2→ Analog configuration F4→ Analog detail
	Group	Group I/O F2→ Group Configuration F4→ Group detail
	Robot	Robot I/O F2→ Robot I/O detail
	UOP	Peripheral device I/O
	SOP	System Operator Panel I/O
	Inter Conect	DI–to–DO connection setting
	Link Device	I/O Link Device List F3→ I/O Unit Model B List F3→ I/O Points Setup
	Flag	Flag screen F2→ Flag detail
6 SETUP	Prog Select	Program selection screen F3→ Detail
	General	General item setting
	Frames	Frame entry F2→ Frame detail
	Macro	Macro entry
	Ref Position	Reference position selection F3→ Reference position setting
	Port Init	Port selection F3→ Port setting
	Ovrd Select	External override setting
	User Alarm	User alarm setting screen
	Error Table	Alarm severity setup screen
	*i*Pendant Setup	*i*Pendant general setup
	BG Logic	Background operation
	Resume Offset	Resume offset setup screen
	Resume Tol.	Resume check screen
	Stroke limit[*d]	Stroke Limit setup

	Space fnct.	Rectangular space F3→ Detail
	Diag Interface	Diagnostics Interface setup screen
	Motion DO[*1]	Motion Group DO setup screen
	Host Comm	Protocol setup F3→ Detail F4→ Selection
	Passwords	Password setup
7 FILE	File	File
	File Memory	File Memory
	Auto Backup	Automatic backup setup screen
8		
9 USER		User
0 — NEXT —		

Fig. A.1 (d) Screen Menu (Page 1)

Screen menu	Screen change menu	Screen
1 SELECT		Program list
2 EDIT		Program edit
3 DATA	Registers	Register
	Position Reg	Position register F4→ Position data information
	String Reg	String register
	Pallet register[*3]	Palletizing register
	KAREL Vars	Karel variable
	KAREL Posns	Karel position variable
4 STATUS	Axis	Robot axis status
	Version ID	Software version
	Stop Signl	Stop Signal Status
	Order File[*a]	Order file
	Exec–hist	Execution history
	Memory	Memory status list F2→ Memory status detail
	Prg Timer	Program Timer Status List F2→ Program Timer Detail
	Sys Timer	System Timer Status

```
│                          ├─Condition              --------Condition Monitor
│                          ├─Program                --------Program status screen
│                          └─Power consumption      --------Power Consumption Monitor screen
├─5 4D GRAPHICS          ──┬─4D Display             --------4D Graphics Display
│                          └─Position Display       --------Current position
├─6 SYSTEM               ──┬─Clock                  --------Calendar
│                          ├─Variables              --------System variable
│                          ├─Servo Param*c          --------Servo parameter
│                          ├─Master/Cal*b           --------Mastering and Calibration
│                          ├─OT Release             --------Over Travel Release
│                          ├─Axis Limits            --------Joint operating area setting
│                          ├─Config                 --------System configuration
│                          └─Motion                 --------Motion Performance
├─7 USER2                ---------------------------------------------
├─8 BROWSER              ──┬─Browser                --------Internet Browser screen    F5→ Robot Homepage
│                          └─Panel setup            --------Operation Panel setup screen
├─9
└─0 — NEXT —
```

Fig. A.1 (e) Screen Menu (Page 2)

Setting

The menu items indicated by * (alphabetical character) in Fig. A.1 (d) and Fig. A.1 (e) are displayed when the corresponding setting is made, as indicated below:

Table A.1 (a) Setting to display the menu item

*	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	Can be displayed depending on the robot type.

Options

The menu items indicated by * (numeral) in Fig. A.1 (a) to Fig. A.1 (e) are displayed when the corresponding option is added, as indicated below:

Table A.1 (b) Option list

*	Option	Specification
1	Multi-motion	A05B-2600-J601
2	Extended axis control	A05B-2600-J518
3	Palletizing	A05B-2600-J500

A.2 LIST OF PROGRAM INSTRUCTIONS

Motion instructions

J
L
C
A

P [ ]
PR [ ]

(feed rate)

FINE
CNT (value)

Additional motion instruction

Wjnt	Wrist joint motion instruction
ACC (value)	Acceleration/deceleration override instruction
Skip, LBL[ ]	Skip instruction
BREAK	Break instruction
Offset	Offset instruction
Offset, PR[ ]	Direct offset condition instruction
INC	Incremental instruction
Tool_offset	Tool offset instruction
Tool_offset, PR[ ]	Direct tool offset instruction
Ind. EV (value) %*1	Independent EV instruction
EV (value) %*1	Simultaneous EV instruction
TB (VALUE) sec …	Time before statement
Skip, LBL[ ] PR[ ]	High-speed Skip instruction
TA (VALUE) sec …	Time after statement
DB (VALUE) mm …	Distance before statement
PTH	Path instruction

Fig. A.2 (a) Motion (option) instruction

Program instruction menu	Program instruction format		Program instruction
1 Register	R [ ] =	R [ ]	Register instruction
		Const	
		DI / O [ ]	
		RI / O [ ]	
		GI / O [ ]	
		SI/O [ ]....	
		UI / O [ ]	
		TIMER [ ]	
		TIMER_OVERFLOW [ ]	
		AR [ ]	
		AI / O [ ]	
		PR [i , j]	
		SR [ ]	
		(....)	
	PR [ ] = ...		Position register instruction
	PR [i , j] = ...		Position register element instruction
	PL [ ][*3] = ...		Palletizing register instruction
	SR [ ] = ...		String register instruction
2 I / O	DO [ ] =	ON	I/O instruction
		OFF	
		PULSE (, Width)	
		R[ ]	
		(…)	
	RO [ ] = ...		
	GO [ ] = ...		
	AO [ ] = ...		
	F [ ] = ...	(…)	
3 IF / SELECT	IF	R [ ] = ...	Comparison

- DI / O [ ] = ...
- RI / O [ ] = ...
- GI / O [ ] = ...
- SI / O [ ] = ...
- UI / O [ ] = ...
- PL [ ] [*3] = ...
- $ (System variable) = ...
- AI / O [ ] = ...
- SR [ ] = ...
- (…)

- SELECT R [ ] = ...
  - R [ ] = ... ------ Selection condition instruction
  - Constant ....
  - AR [ ] = ...
  - SR [ ] = ….

- 4 WAIT
  - WAIT (time) -- Time wait instruction
  - WAIT
    - R [ ] = ... ------ Conditional wait instruction
    - DI / O [ ] = ...
    - RI / O [ ] = ...
    - GI / O [ ] = ...
    - SI / O [ ] = ...
    - UI / O [ ] = ...
    - ERR_NUM = ...
    - $ (System variable) = ...
    - AI / O [ ] = ...
    - (…)
- 5 JMP / LBL
  - LBL [ ] -- Label instruction
  - JMP LBL [ ] -- Jump instruction
- 6 CALL / END
  - CALL (Program) -- Program call instruction
  - END -- Program end instruction

7 Palletizing[*3]	Palletizing – B		Palletizing instruction
	PALLETIZING – BX		Palletizing BX statement
	PALLETIZING – E		Palletizing E statement
	PALLETIZING – EX		Palletizing EX statement
	Palletizing – END		Palletizing end instruction

Fig. A.2 (b) Program instruction menu

Program instruction menu	Program instruction format		Program instruction
8 Miscellaneous	RSR [ ] = ...	ENABLE	RSR statement
		DISABLE	
	UALM [ ]		User alarm instruction
	TIMER [ ] = ...	START	Timer instruction
		STOP	
		RESET	
		(…)	
	OVERRIDE = ...	R [ ]	Override instruction
		Constant	
		AR [ ]	
	! (Remark)		Comment instruction
	Message [ ]		Message instruction
	Parameter	$ (Parameter) = ...	Parameter instruction
		.... = $ (Parameter)	
	MAX_SPEED	JOINT_MAX_ SPEED [ ] = ...	Max speed instruction
		LINEAR_MAX_ SPEED = ….	
	-- (Remark)		Multi-Language Comment instruction
9 Skip	SKIP CONDITION I/O [ ] = ...		Skip condition instruction
10 Payload	PAYLOAD [ ]		Payload setting statement
11 Offset	OFFSET CONDITION PR [ ] = ...		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 Multiple control	RUN (PROGRAM)	Run program statement
13 Program Control	PAUSE	Halt instruction
	ABORT	Abort instruction
	ERR_PROG = (PROGRAM)	Error program instruction
	RESUME_PROG [ ]= (PROGRAM)	Resume program instruction
14 MACRO		Macro instruction

Fig. A.2 (c) Program instruction menu

Program instruction menu	Program instruction format	Program instruction
15 FOR/ENDFOR	FOR ….	FOR instruction
	ENDFOR	ENDFOR instruction
16 Tool_Offset	TOOL_OFFSET CONDITION PR [ ] = ...	Tool offset condition instruction
17 LOCK PREG	LOCK PREG	Position register lock instruction
	UNLOCK PREG	Position register unlock instruction
18 MONITOR/MON.END	MONITOR	Monitor start instruction
	MONITOR END	Monitor end instruction
19 Independent GP*2	Independent GP	Independent motion group Instruction
20 Simultaneous GP*2	Simultaneous GP	Simultaneous motion group instruction
21 STRING	R [ ] = STRLEN SR [ ]	Get string Length
	SR [ ] = SUBSTR SR [ ], R [ ], R [ ]	Subtract string
	R [ ] = FINDSTR SR [ ], SR [ ]	Find string

Fig. A.2 (d) Program instruction menu

Program instruction menu	Program instruction format	Program instruction
22 DIAGNOSE	DIAG_REC[…, ..., …]	Diagnose instruction
	DIAG_REC_SEC[..., …]	Diagnose instruction(in second)

Fig. A.2 (e) Program instruction menu

Option

The items indicated by * (numeral) in Fig. A.2 (a) to A.2 (e) are displayed when the corresponding option is added, as indicated below:

Table A.2 Option list

*	Option	Specification
1	Extended axis control	A05B-2600-J518
2	Multi motion group	A05B-2600-J601
3	Palletizing	A05B-2600-J500

A.3 PROGRAM INSTRUCTIONS

A.3.1 Motion Instructions

Table A.3.1 Motion instructions

Motion format	J L C, A	Enables robot operation for each joint with interpolation. Moves the robot tool linearly. Enables the tool center point of the robot to make a circular motion.
Position variable	P[i : Comment] PR[i : Comment]	Standard variable for storing position data. Register for storing position data.
Feed rate unit	% mm/sec, cm/min, inch/min, deg/sec sec, msec	Specify the rate of a feed rate to the highest feed rate of the robot. Specifies the speed with which the tool center point makes a linear or circular or circle arc motion. Specify the time required during a motion.
Positioning path	FINE CNTn n (0 - 100):	The robot stops at the specified position and starts the next motion. The robot moves gradually from the specified position to the position at which the next motion starts. Degree of gradual motion. The higher the specified number, the more gradual the robot moves.

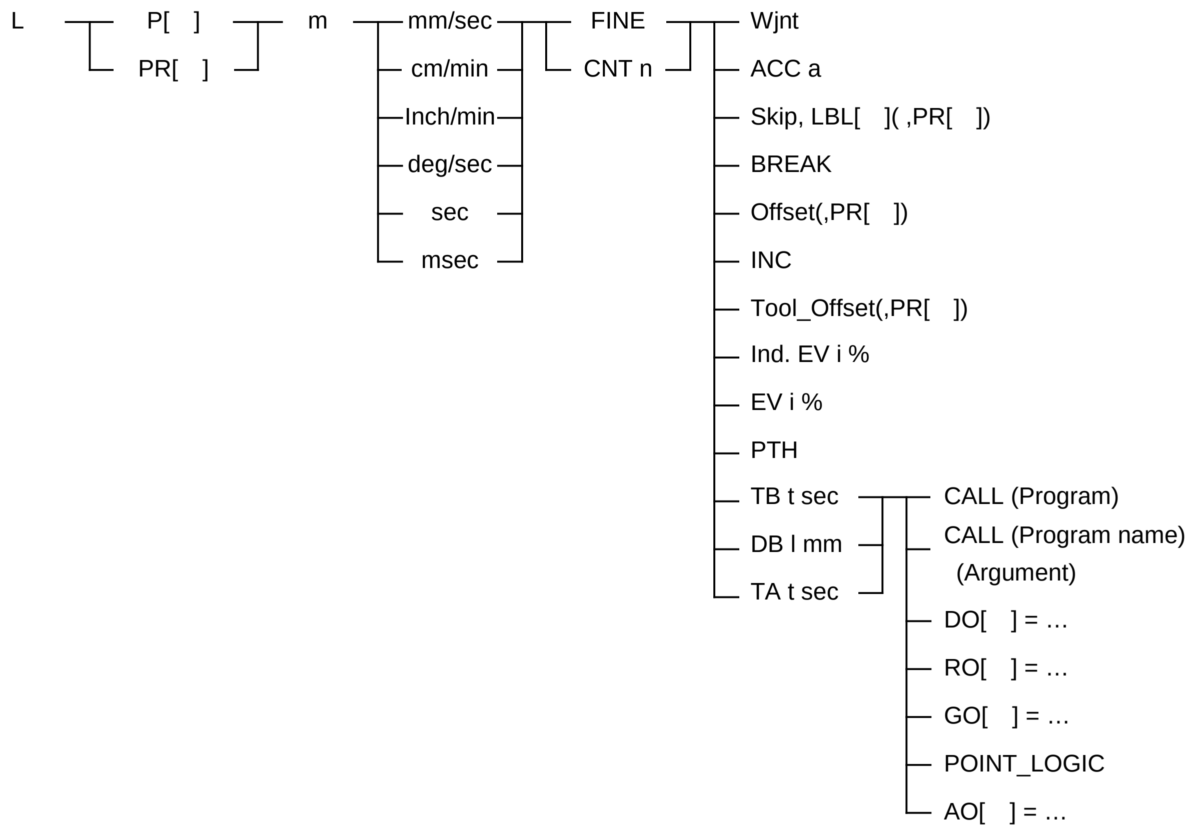
A.3.2 Additional Motion Instructions

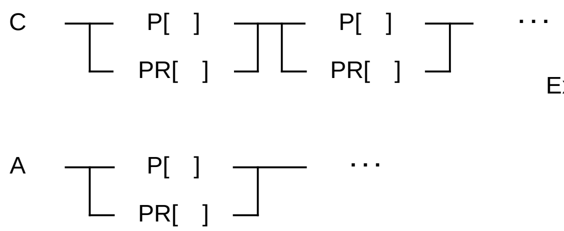
Table A.3.2 Additional motion instructions

Wrist joint motion	Wjnt	On a linear or arc or circle 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[i] Skip, LBL[i], PR[j]	Causes a branch to the specified label when the condition specified in a skip condition instruction is not satisfied. When the condition is satisfied, cancels the motion and executes the next line.
Positional offset	Offset Offset, PR[(GPk:) i]	Makes the robot move to the position where the value specified by the offset condition instruction is added to the positional variable. Makes the robot move to the position where the value specified by the offset condition instruction and the value of position register are added to the positional variable.
Tool offset	Tool_offset Tool_offset,PR[(GPk:)i]	Moves the robot to the position corresponding to the value specified by the tool offset instruction, added to the position variable. Moves the robot to the position corresponding to the position register value, added to the position variable.

Incremental	INC	Makes the robot move to the position where the value of the position variable is added to the current position.
Independent EV	Ind.EV(i)% i = 1 to 100 (%)	Moves the extended axis, independently of the robot motion.
Simultaneous EV	EV(i)% 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.
Before execution	TB t sec <Action> DB l mm <Action> TA t sec <Action>	Before or after the specified time or distance at which the robot motion terminates, the subprogram call or signal output can be executed. t = Execution start time. l = Execution start distance The instruction which can be specified as <Action> are as follows. • Call (Program) • Call (Program) (Argument) • DO [i] = … • RO [i] = … • GO [i] = … • POINT_LOGIC • AO [i] = …
BREAK	BREAK	Even if the positioning path of the motion immediately before a WAIT instruction is CNT, the robot moves toward the taught position until the condition in the WAIT instruction is satisfied.

```
J ─┬─ P[ ]  ─┬─ m ─┬─ %    ─┬─┬─ FINE  ─┬─┬─ ACC a
   └─ PR[ ] ─┘     ├─ sec  ─┤ └─ CNT n ─┘ ├─ Skip, LBL[ ] (,PR[ ])
                   └─ msec ─┘             ├─ BREAK
                                          ├─ Offset(,PR[ ])
                                          ├─ INC
                                          ├─ Tool_Offset(,PR[ ])
                                          ├─ Ind. EV i %
                                          ├─ EV i %
                                          ├─ PTH
                                          ├─ TB t sec ─┬─┬─ CALL (Program)
                                          ├─ DB l mm  ─┤ ├─ CALL (Program name)
                                          └─ TA t sec ─┘ │    (Argument)
                                                         ├─ DO[ ] = …
                                                         ├─ RO[ ] = …
                                                         ├─ GO[ ] = …
                                                         ├─ POINT_LOGIC
                                                         └─ AO[ ] = …
```





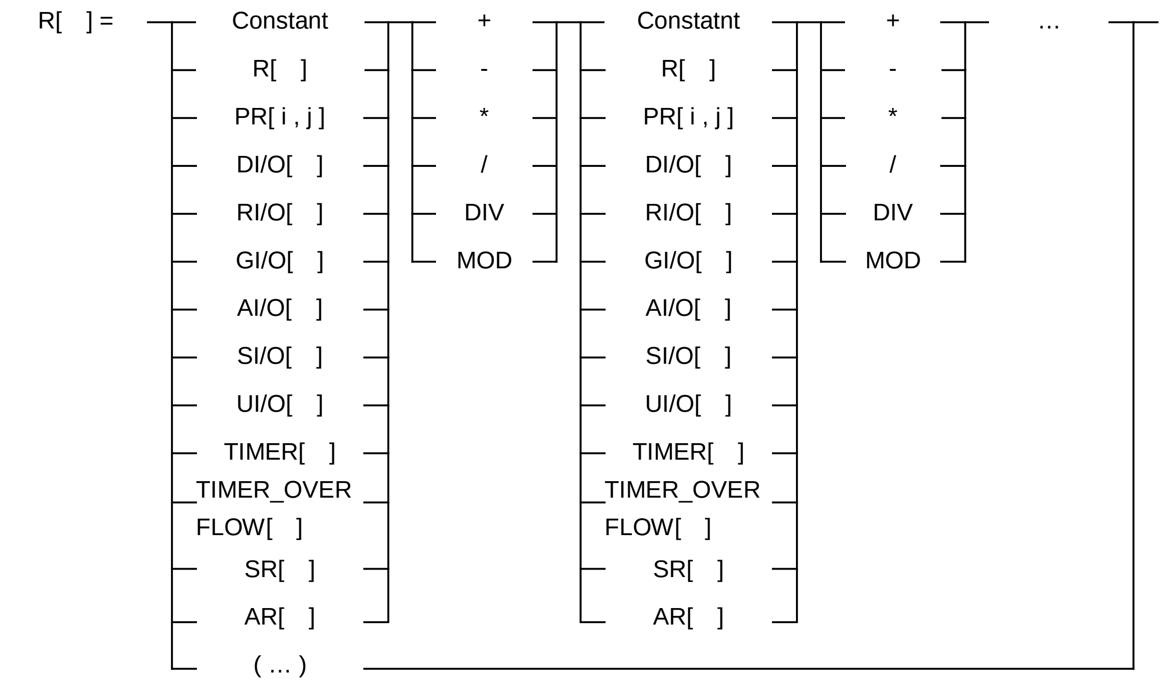
Example
1:J P[1] 100% FINE
2:L P[2:LINE] 500mm/sec CNT100
Wjnt Offset,PR[1]
3:L P[3] 3.5 sec CNT100 INC
4:L P[4] 100cm/min FINE
Skip LBL[100] Wjnt
5:C P[5]
P[6] 300mm/sec CNT 50
6:A P[7] 50mm/sec CNT 100
7:A P[8] 50mm/sec CNT 100
8:A P[9] 50mm/sec FINE

A.3.3 Register and I/O Instructions

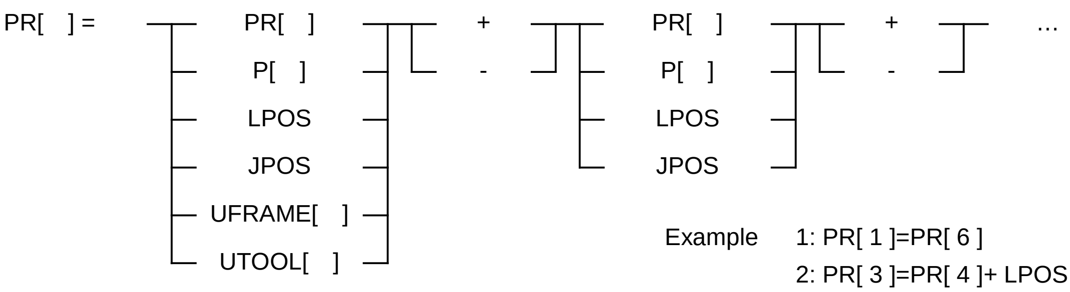
Table A.3.3 Register and I/O instructions

Register	R[i]	i: Register number.
Position register	PR[i] PR[(GPk:) i] PR[(GPk:) i , j]	Register to store a position data. i: Position register number k:Group number Fetches a position data element. j: Number of an element in a position register
String register	SR[i]	Register to store a string data. i: String register number

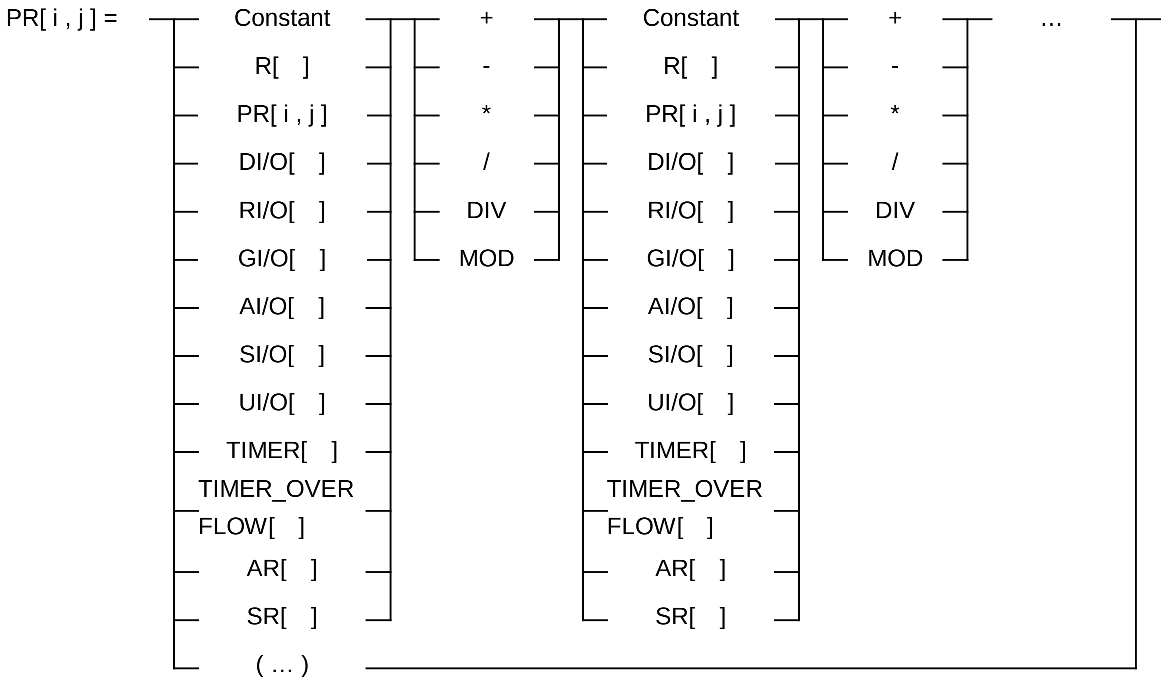
Position data	P[i :comment] Lpos Jpos UFRAME [i] UTOOL [i]	i: Position number. Cartesian coordinates of the current position Joint coordinates of the current position User coordinate system Tool coordinate system
Input/output signal	DI[i], DO[i] RI[i], RO[i] GI[i], GO[i] AI[i], AO[i] UI[i], UO[i] SI[i], SO[i]	(System) digital signals Robot (digital) signals Group signals Analog signals Peripheral signals Operator's panel signals

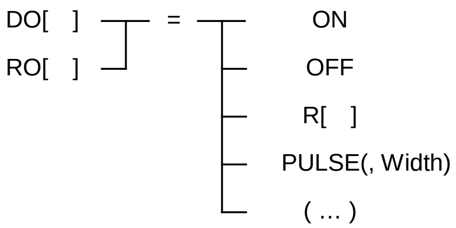


Example
1: R[1]=RI[3]
2: R[3]=DI[4]* PR[1, 2]
3: R[4]=AI[1]

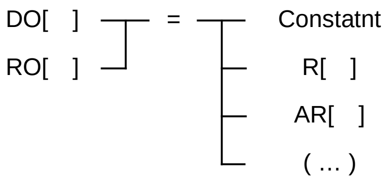


Example
1: PR[1]=PR[6]
2: PR[3]=PR[4]+ LPOS
3: PR[8]=UFRAME[1]
4: PR[9]=UTOOL[2]





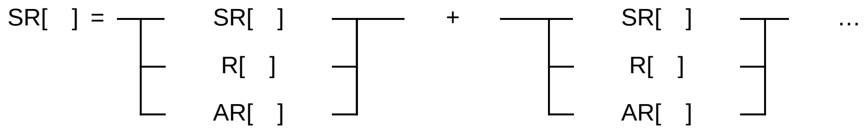
Example
1: DO[1] = ON
2: RO[3] = PULSE 1.0sec
3: RO[4] = R[1]
4: GO[9] = R[2]
5: AO[10] = 12.5

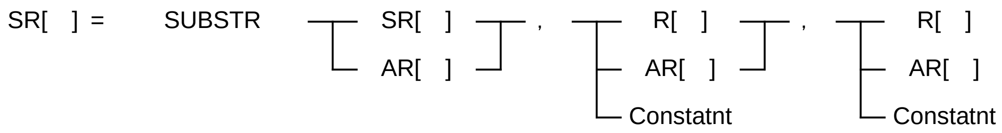


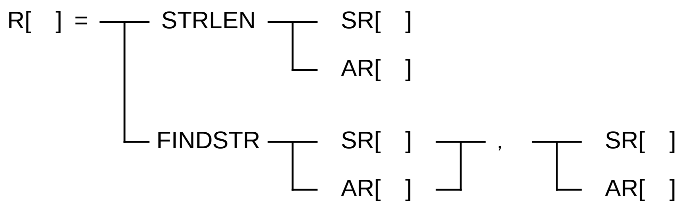
A.3.4 String Register, String Instructions

Table A.3.4 String register, string instructions

String register	SR[i]	Register to store a string data. i: String register number
Find string	R[i] = FINDSTR SR[A] SR[B]	Finds the string specified by SR[B] within the string specified by SR[A], and returns the index.
Get string	SR[i] = SUBSTR SR[j], R[A], R[B]	Returns a sub string from the string specified a string register based on the start point specified by R[A] and the length specified R[B].
Count string length	R[i] = STRLEN SR[j]	Stores the string length of the string specified a string register.





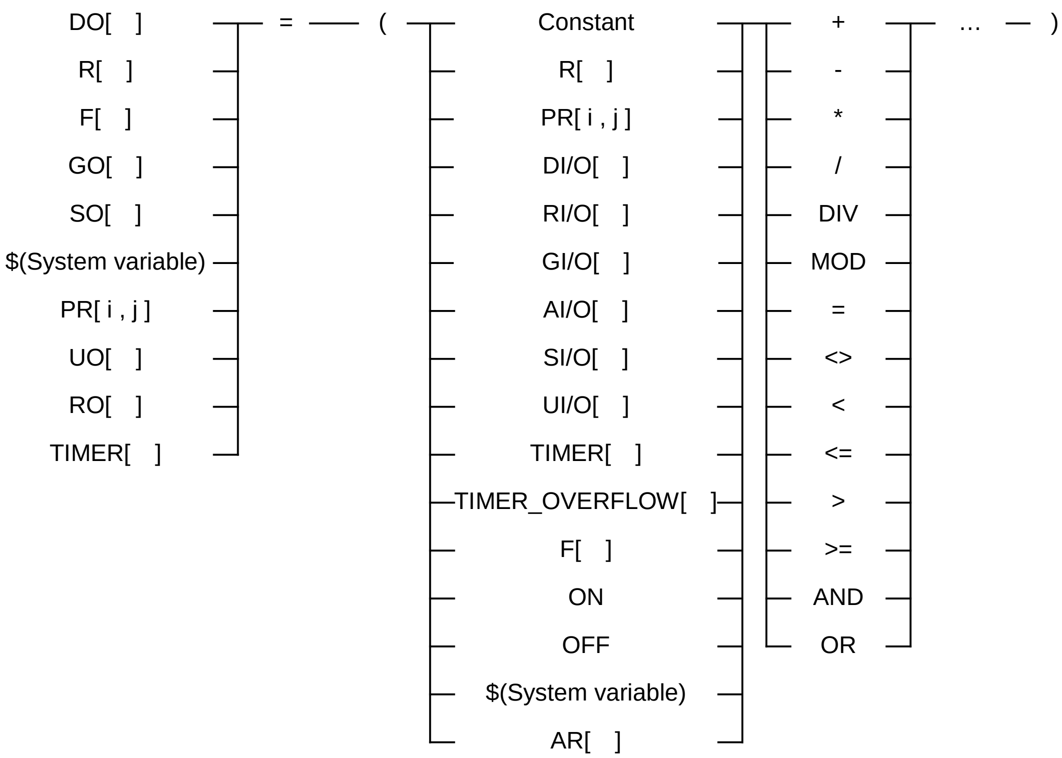


Example 1: SR[1] = SUBSTR SR[1] ,R[2], R[3]
2: R[4] = STRLEN SR[2]
3: R[5] = FINDSTR SR[3], SR[4]

A.3.5 Mixed Logic Instructions

Table A.3.5 Mixed logic instructions

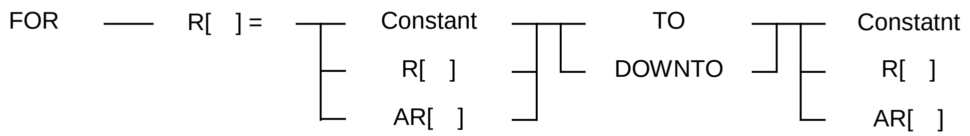
Mixed logic	… = (…) IF (…) branch Wait (…)	Allows to combine various data and various operator in assignment instruction, comparison condition instruction, and wait instruction.



A.3.6 FOR/ENDFOR Instructions

Table A.3.6 FOR/ENDFOR instructions

FOR	FOR (Loop Counter) = (Initial value) TO / DOWNTO (Target value)	Repeats a loop within FOR and ENDFOR statements until the loop counter started from initial value reaches to the target value.
ENDFOR	ENDFOR	Indicates the end of the loop.

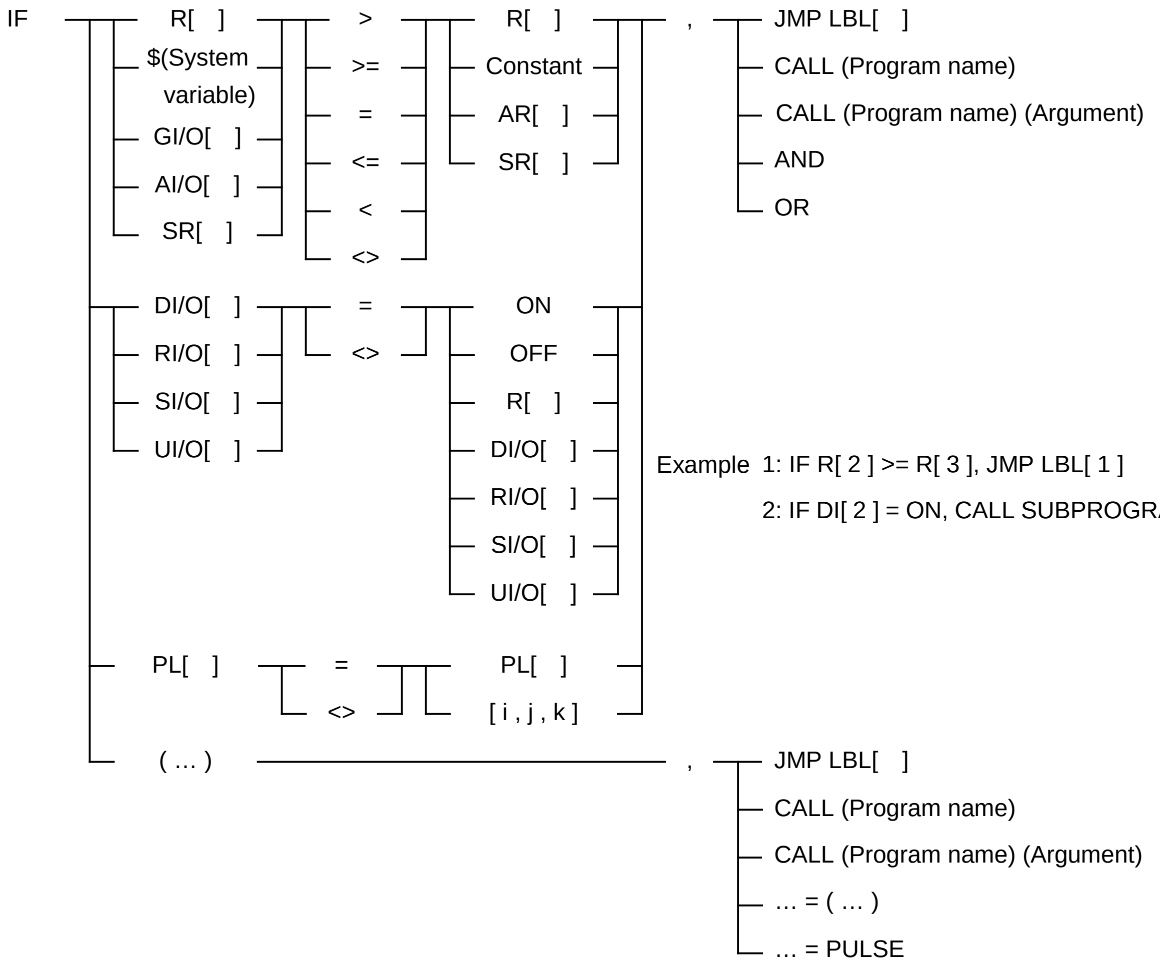


ENDFOR

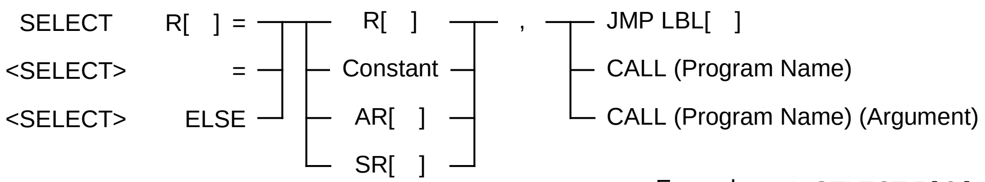
A.3.7 Conditional Branch Instructions

Table A.3.7 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], JMP LBL[1]
2: IF DI[2] = ON, CALL SUBPROGRAM

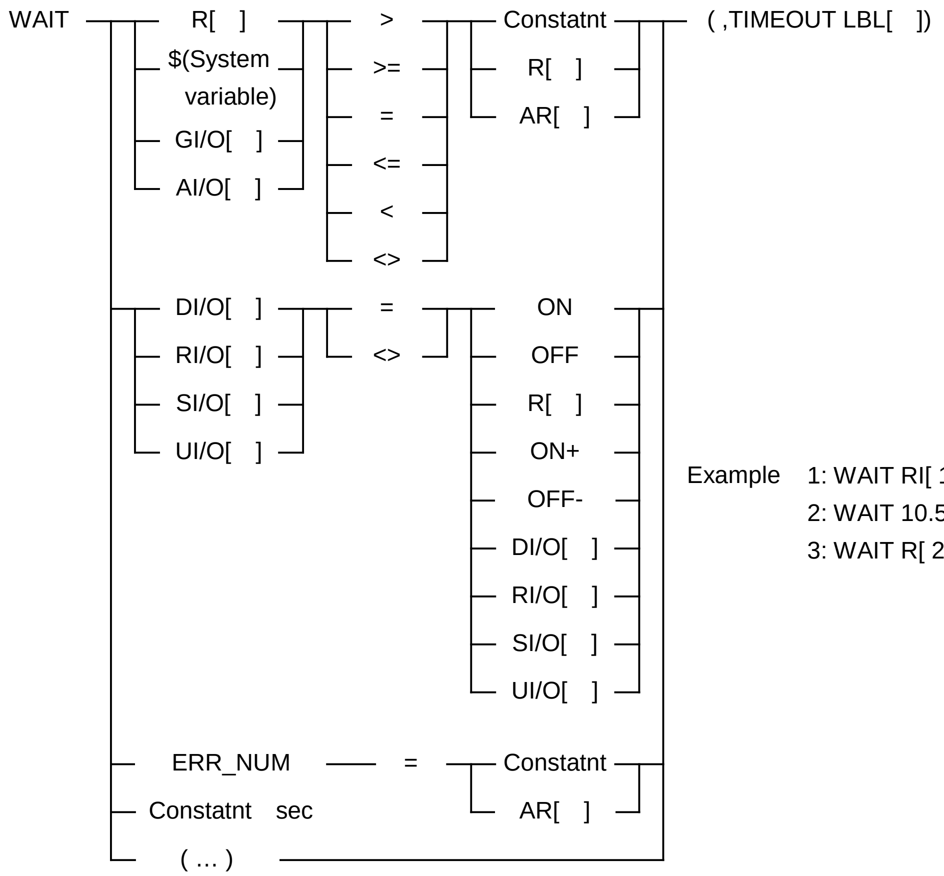


Example 1: SELECT R[2] = 1, JMP LBL[1]
2: =2, JMP LBL[2]
3: =3, JMP LBL[3]
4: ELSE CALL MAINPROG

A.3.8 Wait Instruction

Table A.3.8 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 RI[1]=ON
2: WAIT 10.5sec
3: WAIT R[2], TIMEOUT LBL[1]

A.3.9 Unconditional Branch Instructions

Table A.3.9 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.

CALL (Program Name) (Argument)

END

JMP LBL[ ]

LBL[ ]

Example 1: LBL[1:HANDCLOSE]
2: JMP LBL[2]
3: CALL PROGRAM1

A.3.10 Program Control Instructions

Table A.3.10 Program control instructions

Halt	PAUSE	Halts a program.
Abort	ABORT	Aborts a program.
Error Program	ERROR_PROG	Program which is started when an alarm occurs. (This instruction is available in only dispense tool.)
Resume Program	RESUME_PROG	Program which is started when an alarm is released. (This instruction is available only in Automatic error recovery function (J924).)

Example 1: PAUSE
2: ABORT
3: ERR_PROG = TOOL_RET
4: RESUME_PROG = TOOL_PRE

A.3.11 Other Instructions

Table A.3.11 Other instructions

RSR	RSR[I]	Enables or disables RSR signals.
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.
Multi-language Comment	--(comment)	Insert a comment in a program. It is possible to input comment to each language independently.
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 motion statements in the program.

```
RSR[   ]      =  ┬─ ENABLE
                 └─ DISABLE
UALM[   ]     =

TIMER[   ]    =  ┬─ START
                 ├─ STOP
                 ├─ RESET
                 └─ (…)

OVERRIDE      =  ┬─ Constant
                 ├─ R[   ]
                 └─ AR[   ]

!(Comment)

MESSAGE[ (Message) ]

$(System variable)  =  ┬─ Constant
                       ├─ R[   ]
                       ├─ PR[   ]
                       └─ AR[   ]

R[   ]  ─┬─  =  $(System variable)
PR[   ] ─┘

JOINT_MAX_SPEED[   ]  =  ┬─ Constant
                         ├─ R[   ]
                         └─ AR[   ]

LINEAR_MAX_SPEED      =  ┬─ Constant
                         ├─ R[   ]
                         └─ AR[   ]

--(Comment)
```

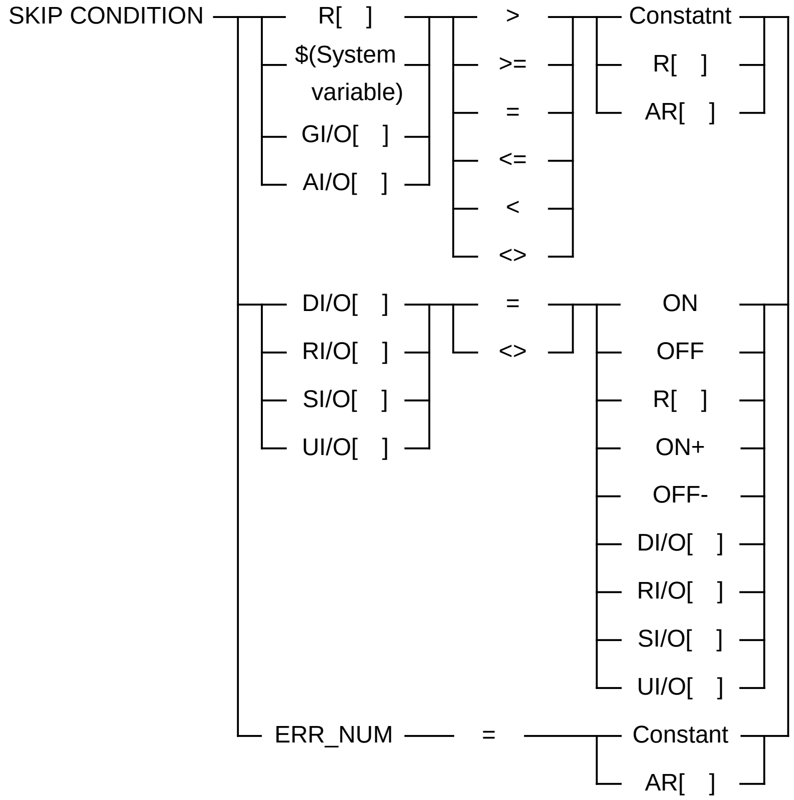
Example

```
 1: RSR[ 2 ] = ENABLE
 2: UALM[ 3 ]
 3: TIMER[ 1 ] = START
 4: OVERRIDE = 50%
 5: !START of STEP 2
 6: MESSAGE[EXECUTE STEP 1]
 7: $NRM_TURN = 1
 8: R[1] = $DEFPULSE
 9: JOINT_MAX_SPEED[2] = R[1]
10: --END OF PROGRAM
```

A.3.12 Skip and Offset Condition Instruction

Table A.3.12 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[ ])

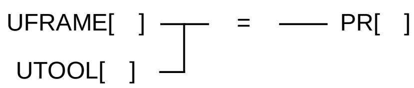
TOOL_OFFSET CONDITION PR[ ] (, UTOOL[ ])

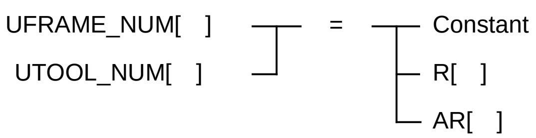
Example 1: SKIP CONDITION DI[1] = ON
2: SKIP CONDITION RI[2] <> DI[3]
3: OFFSET CONDITION PR[1], UFRAME[1]
4: TOOL_OFFSET CONDITION PR[2], UTOOL[1]

A.3.13 Frame Setup Instruction

Table A.3.13 Frame setup instruction

User frame User frame selection	UFRAME[i] UFRAME_NUM	User frame The number of current user frame
Tool frame Tool frame selection	UTOOL[i] UTOOL_NUM	Tool frame The number of current tool frame





Example 1:UTOOL[3] = PR[1]
2:UFRAME_NUM = 3

A.3.14 Macro Instruction

Table A.3.14 Macro instruction

Macro	(macro-instruction)	Executes a program defined on the macro instruction setting screen.

Example 1: HAND1 OPEN

2: HAND2 CLOSE

A.3.15 Multiaxis Control Instructions

Table A.3.15 Multiaxis Control Instructions

Program execution	RUN	Starts execution of a specified program in another motion group.

RUN(Program name)

Example

PROGRAM1	PROGRAM2
1: R[1] = 0	1: J P[3] 100% FINE
2: RUN PROGRAM2	2: J P[4] 100% FINE
3: J P[1] 100% FINE	3: J P[5] 100% FINE
4: J P[2] 100% FINE	4: J P[6] 100% FINE
5: WAIT R[1]= 1	5: R[1]= 1
Group Mask [1,*,*,*,*,*,*,*]	Group Mask[*,1,*,*,*,*,*,*]

A.3.16 Position Register Look-ahead Execution Instruction

Table A.3.16 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[ 10 ]
 3:    PR[ 2 ] = PR[ 11 ]
 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.3.17 Status Monitoring Instructions

Table A.3.17 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.3.18 Motion Group Instructions

Table A.3.18 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
  : GP2 J  P[ 2 ] 70% CNT50
```

A.3.19 Diagnose Instruction

Table A.3.19 Diagnose instruction

Diagnostic record	DIAG_REC[Arg1, Arg2, Arg3]	Records the data required for Robot Condition Analysis in ROBOGUIDE DiagnosticsPRO.

```
DIAG_REC [ ─┬─ Constant ─┬─ , ─┬─ Constant ─┬─ , ─┬─ Constant ─┬─ ]
            ├─ R[ ]     ─┤     ├─ R[ ]     ─┤     ├─ R[ ]     ─┤
            └─ AR[ ]    ─┘     └─ AR[ ]    ─┘     └─ AR[ ]    ─┘
```

Example
```
1:    DIAG_REC[1, 1, 500]
2: J  P[ 1 ] 100% FINE
3: J  P[ 2 ] 100% FINE
```

A.3.20 Palletizing Instructions (Software Option:J500)

Table A.3.20 Palletizing instructions

Palletizing	PALLETIZING-B(BX,E,EX)_i	Calculates palletizing. i: Palletizing number
Palletizing end	PALLETIZING-END_i	Increases or decreases a palletizing counter in the palletizing register. i: Palletizing number
Palletizing motion	L PAL_i[A(R)-j] 300mm/s FINE	Specifies the palletizing position. i: Palletizing number j: Number of the approach (retraction) point
Palletizing register	PL[i]	Register to manage the position of the stack point in palletizing. i: Palletizing register number

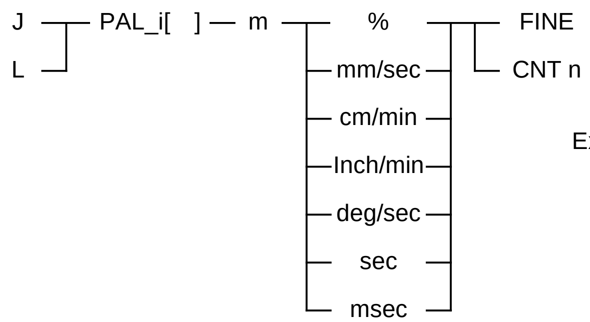
PALLETIZING-B_i

PALLETIZING-BX_i

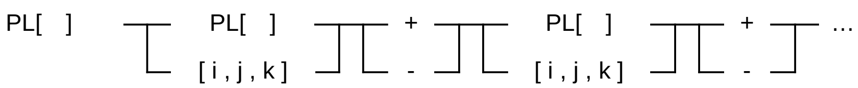
PALLETIZING-E_i

PALLETIZING-EX_i

PALLETIZING-END_i



Example
1: PALLETIZING-EX_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



Example
1: PL[1] = PL[3] + [1 , 2 , 1]
2: PL[2] = [1 , 2 , 1] + [1 , R[1], *]

B SPECIAL OPERATION

This appendix summaries items necessary for using this model. It may also be used as an index.

Contents of this appendix

B.1 START MODE

B.1.1 Start Up Methods

Robot controller has the following four start up methods (start mode):

Initial start

When the controller 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 controller 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 up the robot. From the menu displayed by pressing [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 failure recovery function 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 failure recovery function 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 failure recovery function is enabled. When the controller starts, the program runs and output signals are restored to the state existing prior to the last power-down. Once the hot start has been completed, the robot can be operated.

The cold start or the hot start is started in usually operation. Which mode is used depends on whether the hot start is enable or disable.
The initial start and the controlled start will be used when maintaining it. These modes will not be used in usually operation.

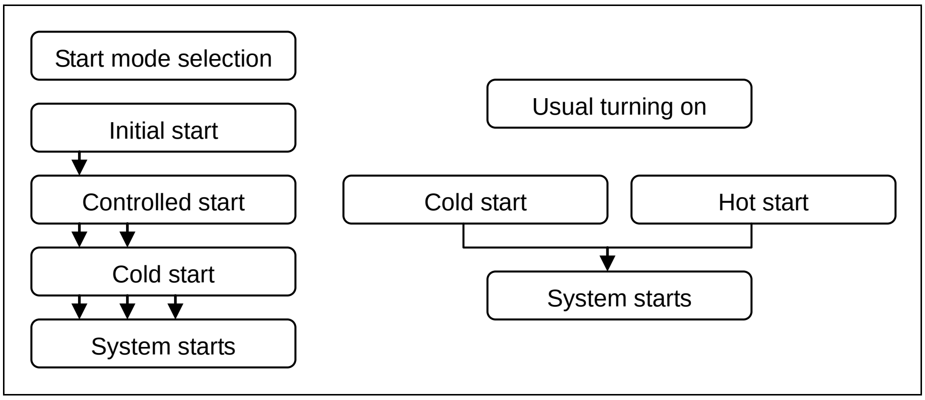


Fig. B.1.1 Start mode

B.1.2 Initial Start

When the controller is started in the initial start mode, all programs are deleted, and all the settings are reset to their standard values.
Once the initial start has been completed, a controlled start is performed automatically.

> ⚠ **CAUTION**
> At an initial start, programs and all data including settings will be lost. The factory-set mastering data is also erased. The initial start should be made only when the main printed circuit board or software is replaced. Before performing an initial start, therefore, make a backup copy of the necessary programs and system files.

Procedure B-1 Initial start

Step

1 With the F1 key and the F5 key held down on the teach pendant, set the power breaker on the controller to ON. BOOT MONITOR screen will be displayed.

```
*** BOOT MONITOR ***
Base version V8.10P/01   [Release 3]
 ******* BMON MENU *******
  1. Configuration menu
  2. All software installation(MC:)
  3. INIT start
  4. Controller backup/restore
  5. Hardware diagnosis
  6. Maintenance
  7. All software installation(Ethernet)
  8. All software installation(USB)

  Select :
```

2 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 will be displayed.

B.1.3 Controlled Start

When the controller 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 [FCTN] key on the controlled start menu. A menu will be displayed. 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:

Tool setup screens

The initial setup depending on the application can be made.

Software version Screen

The software edition is displayed.

System variables screen

System variables can be set. Even a system variable which cannot normally be changed (R0) can be changed.

File screen

A program or system file can be saved and read. The system file can be read only from the controlled start menu.

On the file screen of the controlled start menu, F4 is displayed as [RESTORE]. When the F4 key is pressed, all files are read automatically. To switch F4 to [BACKUP] as on other file screens, press [FCTN] key. A menu will be displayed. From that menu, select RESTORE/BACKUP.

Alarm history screen

The alarm history is displayed.

Port Init screen

A serial port is set.

Memory screen

The memory status is displayed.

MAINTENANCE

Robot setting such as mount type and payload mode can be changed. Extended Axis can be set.

> ⚠ **CAUTION**
> Normally, Robot Maintenance is executed for initial setting, and all motion parameters of target robot and extend axis will return to default. Therefore, path, cycle time, and joint axis limit etc. might change by executing Robot Maintenance, even if you didn’t change the setting.

Max. number setting screen

The numbers of resistor, macro, and user alarm can be changed.

Password setting screen

In case that each setting is restricted according to password level, the restriction can be released by entering the password in this screen.

Host communication setting screen

Communication setting can be made in this screen. In order to load some files by communication at controlled start, set up communication in this screen.

Procedure B-2 Controlled start

Step

1 With the [PREV] key and the [NEXT] key held down on the teach pendant, set the power breaker on the controller to ON.

```
System version: V8.1035        2/10/2012

---------- CONFIGURATION MENU ----------

  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 will be displayed .

```
Tool Setup
                                          1/6
     1 F Number                       F00000
     2 KAREL Prog in select menu      YES
     3 Remote device:                 UserPanel
     4 Intrinsically safe TP:         NO


   [ TYPE ]                     HELP
```

⚠ CAUTION

Please verify the rest of FROM is more than 600KB when you need to do the controlled start. You can verify the available FROM in memory status menu. [select menu →0.Next→4.status→memory]. (7.12 MEMORY USE STATUS DISPLAY)
If the rest of FROM is less than 600KB, please remove the unnecessary data from FR: or FRA: device to make room in FROM before you do the controlled start.
Also you cannot select controlled start in configuration menu if the rest of FROM is less than 600KB.

3 To operate the robot, a cold start must be performed. To do this, press [FCTN] key. A menu will be displayed. From that menu, select 1 START (COLD). A cold start is performed.

B.1.4 Cold Start

The cold start mode is used when normal power-up is performed while power failure recovery function 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 feed rate 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 setting of the power failure recovery function.

Procedure B-3 Cold start

Condition

■ Hot start is set to disable.

Step

1 Turn on the power breaker on the controller to ON. The following screen is displayed after the system stars by cold start.

```
UTILITIES Hints

                 HandlingTool
         V8.10P/01              7DC1/01


      Copyright 2012, All rights Reserved
               FANUC CORPORATION
       FANUC Robotics America Corporation
     Licensed Software: Your use constitutes
     your acceptance. This product protected
     by several U.S. patents.


   [ TYPE ] LICENSE  PATENTS           HELP
```

Procedure B-4 Cold start

Condition

■ Hot start is set to enable.

Step

1 With the [PREV] key and the [NEXT] key held down on the teach pendant, set the power breaker on the controller to ON. The configuration menu will be displayed.

```
System version: V8.1035        2/10/2012

---------- CONFIGURATION MENU ----------

  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

                  HandlingTool
         V8.10P/01              7DC1/01


      Copyright 2012, All rights Reserved
               FANUC CORPORATION
        FANUC Robotics America Corporation
     Licensed Software: Your use constitutes
     your acceptance. This product protected
     by several U.S. patents.



   [ TYPE ] LICENSE  PATENTS           HELP
```

B.1.5 Hot Start

The hot start mode is used when normal power-up is performed while power restoration is enabled. Upon a hot start, the following is performed:

- Each output signal of digital I/O, analog I/O, robot I/O, and group I/O is set in the same manner as it was prior to the last power-down.
- The program runs in the same way as it did prior to the last power-down. If the program was running up until the last power-down, the program enters the pause state.
- The feed rate override, manual feed coordinate system, and machine lock are set in the same manner as they were prior to the last power-down.

> ⚠ **CAUTION**
> When a hot start is performed in the following state, each output signal of the digital I/O, analog I/O, robot I/O, and group I/O is turned off or set to 0:
> - When the I/O allocation is changed
> - When an I/O unit is mounted or removed
> - When the number of signals is changed on the I/O Link screen

Procedure B-5 Hot start

Condition

■ Hot start must be set to enable.

Step

1 Turn on the power breaker on the controller to ON. The screen which was being displayed at power off will be displayed on the screen of the teach pendant after a few seconds.

B.2 MASTERING

Mastering associates the angle of each axis of the robot with the pulse count of the Pulsecoder 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 Pulsecoders 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 pulse counts are lost for some reason such as a drop in the voltage of the backup battery for the pulse counts backup in the mechanical unit or exchange of Pulsecoder.
- The pulse counts do not indicate the angles of the axes because the mechanical unit was hit bumped, etc.

> ⚠ **CAUTION**
> The robot data including mastering data and the Pulsecoder data are maintained independently by backup batteries. If the batteries go empty, data is lost. To prevent this, replace both batteries periodically. When the battery voltage drops, an alarm `BLAL' notifies the user.

There are six types of mastering as listed below.
If software version 7DC2(V8.20P) or former software is installed, "Quick Mastering for Single Axis" has not been supported.

Table B.2 Mastering types

Type of mastering	Explanation
Fixture position mastering	Mastering is performed using a special jig. Fixture position 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.
Quick mastering for single axis	"Quick mastering for single axis" performs "Quick mastering" for selected axes. (Reference points must be set in advance.)
Single axis mastering	"Single axis mastering" performs mastering for selected axes.
Setting mastering data	Mastering data is set in mastering counters directly.

> ⚠ **CAUTION**
> After the robot is installed, the quick mastering reference points should be stored in case the factory-adjusted settings are needed in mastering in the future.

After mastering, be sure to perform positioning (calibration). Positioning means that the controller reads the current pulse counts and recognizes the current position.

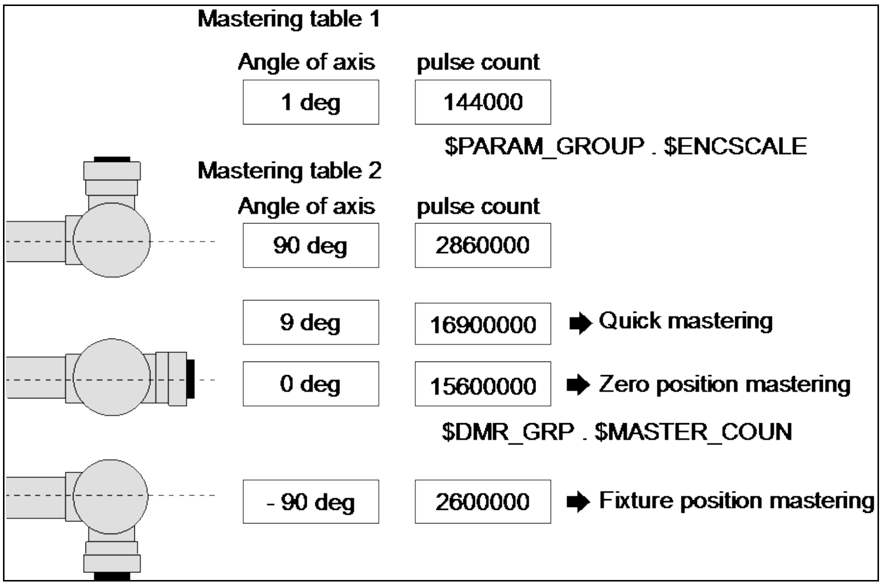


Fig. B.2 Mastering

The current position of the robot is determined by the following data:

- Pulse count per degree. (See mastering table 1.) This value is defined in system variable $PARAM_GROUP.$ENCSCALE.
- Pulse count at the zero-degree position. (See mastering table 2.) This data is stored in $DMR_GRP.$MASTER_COUN by mastering.
  - In jig mastering, the pulse count at the jig position is received and converted to mastering data.
  - In quick mastering, the pulse count at the quick mastering reference position defined by the user is received and converted to mastering data.
- Current pulse count. The current pulse count is received from the Pulsecoder by calibration.

Mastering and calibration are performed on the Master/Cal screen [6 SYSTEM, Master/Cal].

<u>Mastering when Gravity Compensation is enabled</u>

When Gravity Compensation is enabled, the procedure of mastering becomes special.

Gravity Compensation is enabled in the following conditions.
(A) Gravity Compensation option (A05B-2600-J649) is installed and it is enabled.
(B) The robot model includes Gravity Compensation (robot arm weight only) by default

For the case (A), refer to a chapter “MASTERING” of “GRAVITY COMPENSATION” in “Optional Function OPERATOR’S MANUAL” (B-83284EN-2) before performing mastering.
In addition, when you start mastering, popup screen such as the following will be displayed.

```
Gravity Compensation is
enabled.
Confirm you have set payload
data correctly.

Execute mastering?
      [No]       Yes
```

If you select “Yes”, mastering will be executed.
If you select “No”, mastering will not be executed and return back to the previous screen.

For the case (B), you can perform mastering with the usual procedure, but the following message may be displayed at the time of Single Axis Masteing or Quick Mastering For Single Axis.

```
Can’t execute Mastering!

Because Gravity Compensation
is enabled, you must master
all at once the robot axes
which need to be mastered.

      [Close]
```

In this case, select all the robot axes that has not been mastered, and perform Single Axis Mastering or Quick Mastering For Single Axis all at once.
If you cannot move the robot accurately to reference positions of all the selected robot axes due to an interference etc, perform a tentative mastering at an arbitrary position first.
For tentative mastering, you can also perform Mastering at Zero-degrees Positions.
After tentative mastering, perform Single Axis Mastering or Quick Mastering For Single Axis accurately at the reference position for each axis.

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_ENB is automatically set to 0 and then the Master/Cal screen cannot 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 Fixture Position Mastering

Fixture position mastering is performed at the factory using a special jig. This mastering is performed at the mastering position set beforehand.
With this mastering, the accurate mastering can be performed by using the special jig.
This mastering is usually unnecessary to perform it in normal operation because this is used at shipment.
For details of jig mastering, refer to the mechanical manual.

Procedure B-6 Fixture position mastering

Condition

■ System variable $MASTER_ENB must be set to 1 or 2.

```
SYSTEM Variables

      258 $MASTER_ENB      1
```

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select ”0 — NEXT —” and then select ”6 SYSTEM”.
3 Press F1, [TYPE]. The screen change menu will be displayed.
4 Select “Master/Cal” on the screen change menu. The positioning screen will be displayed.

```
SYSTEM Master/Cal
                              TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE

      Press 'ENTER' or number key to select.

   [ TYPE ]          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.

NOTE

Brake control can be released by setting the system variables as follows:
$PARAM_GROUP.$SV_OFF_ALL : FALSE
$PARAM_GROUP.$SV_OFF_ENB[*] : FALSE (for all axes)
After changing the system variables, cycle power of the controller.

6 Select “1 FIXTURE POSITION MASTER” and press the F4, YES key. Mastering data is set.

```
SYSTEM Master/Cal
                              TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE

      Press 'ENTER' or number key to select.

    Master at master position? [NO]

   [ TYPE ]                    YES      NO
```

```
SYSTEM Master/Cal
                              TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE
     Robot Mastered! Mastering Data:
       <-3105333> <-13216881> <22995280>
       <-1354153> <0> <0>

   [ TYPE ]          RES_PCA          DONE
```

7 Select “7 CALIBRATE” and press the F4, YES key. Calibration is performed.

```
SYSTEM Master/Cal
                             TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE

      Press 'ENTER' or number key to select.

    Calibrate? [NO]

   [ TYPE ]                    YES       NO
```

```
SYSTEM Master/Cal
                             TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE
     Robot Calibrated! Cur Jnt Ang(deg):
       <   0.0000> <   0.0000> <   0.0000>
       <   0.0000> <   0.0000> <   0.0000>

   [ TYPE ]          RES_PCA             DONE
```

8 Press F5, DONE, after mastering.
9 Alternatively, to perform positioning, turn the power off, then turn it on again. Calibration is performed whenever the power is turned on.

B.2.2 Mastering at the Zero-degree Positions

Mastering at the zero-degree positions is performed for the robot with its all axes at the zero-degree positions. On each axis of the robot, a zero-degree position mark is attached. Using these marks as a reference, move the robot by jog feed to the zero-degree positions for all axes.
Mastering at the zero-degree positions cannot be performed as accurate by as other types of mastering because it relies on visual adjustment. Perform mastering at the zero-degree positions only as an emergency measure.
For details of mastering at the zero-degree positions, refer to the mechanical manual.

Procedure B-7 Mastering at the zero-degree positions

Condition

■ System variable $MASTER_ENB must be set to 1 or 2.

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select ”0 ─ NEXT ─” and then select ”6 SYSTEM”.
3 Press F1, [TYPE]. The screen change menu will be displayed.
4 Select “Master/Cal” on the screen change menu. The Master/Cal screen will be displayed.

```
SYSTEM Master/Cal
                             TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE

      Press 'ENTER' or number key to select.

  [ TYPE ]         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, YES key. Mastering data is set.

```
SYSTEM Master/Cal
                             TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE

      Press 'ENTER' or number key to select.

    Master at zero position? [NO]
  [ TYPE ]                  YES      NO
```

```
SYSTEM Master/Cal
                             TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE
     Robot Mastered! Mastering Data:
       <0> <11808249> <38768756>
       <9878638> <122000309> <0>

  [ TYPE ]         RES_PCA          DONE
```

7 Select "7 CALIBRATE" and press the F4, YES key. Calibration is performed.

```
SYSTEM Master/Cal
                             TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE

      Press 'ENTER' or number key to select.

    Calibrate? [NO]

  [ TYPE ]                  YES      NO
```

```
SYSTEM Master/Cal
                              TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE
     Robot Calibrated! Cur Jnt Ang(deg):
       <   0.0000> <   0.0000> <   0.0000>
       <   0.0000> <   0.0000> <   0.0000>

   [ TYPE ]           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.

- If mastering data is lost due to the empty of the backup battery for the Pulsecoder, quick mastering can be used.
- When the Pulsecoder is replaced or when mastering data in the robot controller is lost, quick mastering cannot be used.

The pulse count is calculated from “number of revolutions” and “Pulsecoder value within one revolution” of the motor. Quick mastering uses the fact that “Pulsecoder value within one revolution” is kept even if “number of revolutions” is lost due to an empty of the battery.

To perform quick mastering, a reference point set after mastering is necessary (→ reference point setting). The reference point is factory-set to the zero position.

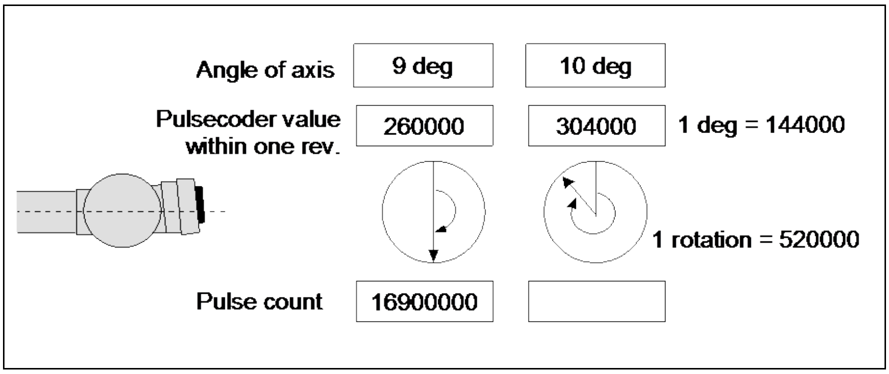


Fig. B.2.3 Quick mastering

When you execute quick mastering, you need to move the robot to the reference position. The misalignment form the reference position should be within half revolution of the motor. For details of quick mastering, refer to the mechanical manual.

> ⚠ **CAUTION**
> If the robot is installed in such a way that the robot cannot be set to the 0 degree position, which is the initial reference position of quick mastering, the reference position of quick mastering should be stored after the installation. This must be done to store the factory-set mastering setting, providing for future quick 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. Press [MENU] key. The screen menu will be displayed.
2. Select ”0 ─ NEXT ─” and then select ”6 SYSTEM”.
3. Press F1, [TYPE]. The screen change menu will be displayed.
4. Select “Master/Cal” on the screen change menu. The Master/Cal screen will be displayed.

```
SYSTEM Master/Cal
                             TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE

      Press 'ENTER' or number key to select.


 [ TYPE ]           RES_PCA           DONE
```

5. Jog the robot to the quick mastering position (reference position). If it is necessary, turn off the brake control.
6. Select “3 QUICK MASTER” and press the F4, YES key. Mastering data is set.

```
SYSTEM Master/Cal
                             TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE

      Press 'ENTER' or number key to select.

    Quick master? [NO]
 [ TYPE ]                     YES      NO
```

```
SYSTEM Master/Cal
                             TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE
     Robot Mastered! Mastering Data:
       <0> <11808249> <38768756>
       <9878638> <122000309> <0>


 [ TYPE ]           RES_PCA           DONE
```

7 Select “7 CALIBRATE” and press the F4, YES key. Calibration will be performed.

```
SYSTEM Master/Cal
                             TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE

      Press 'ENTER' or number key to select.

    Calibrate? [NO]

   [ TYPE ]                YES      NO
```

```
SYSTEM Master/Cal
                             TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE
     Robot Calibrated! Cur Jnt Ang(deg):
       <   0.0000> <   0.0000> <   0.0000>
       <   0.0000> <   0.0000> <   0.0000>

   [ TYPE ]       RES_PCA           DONE
```

8 Press F5, DONE after mastering.

Procedure B-9 Setting reference position for quick mastering (If the robot is installed in such a way that the robot cannot be set to the 0 degree position)

> ⚠ **CAUTION**
> This operation cannot be executed if the mastering data is lost because of mechanical disassembly or maintenance. If that is the case, jig mastering or zero-degree positions mastering should be executed to restore the mastering data.

Condition

- System variable $MASTER_ENB must be set to 1 or 2.

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select ”0 ─ NEXT ─” and then select ”6 SYSTEM”.
3 Press F1, [TYPE]. The screen change menu will be displayed.
4 Select “Master/Cal” on the screen change menu. The Master/Cal screen will be displayed.

```
SYSTEM Master/Cal
                              TORQUE = [ON  ]
    1 FIXTURE POSITION MASTER
    2 ZERO POSITION MASTER
    3 QUICK MASTER
    4 QUICK MASTER FOR SINGLE AXIS
    5 SINGLE AXIS MASTER
    6 SET QUICK MASTER REF
    7 CALIBRATE

     Press 'ENTER' or number key to select.

  [ TYPE ]        RES_PCA         DONE
```

5 Move the robot by jog feed to the quick mastering reference position. Set brake control to off, if necessary.

6 Select “6 SET QUICK MASTER REF” and press the F4, YES key. The reference position for quick mastering are stored in memory.

```
SYSTEM Master/Cal
                              TORQUE = [ON  ]
    1 FIXTURE POSITION MASTER
    2 ZERO POSITION MASTER
    3 QUICK MASTER
    4 QUICK MASTER FOR SINGLE AXIS
    5 SINGLE AXIS MASTER
    6 SET QUICK MASTER REF
    7 CALIBRATE

     Press 'ENTER' or number key to select.

  [ TYPE ]        RES_PCA         DONE
```

B.2.4 Quick Mastering For Single Axis

Quick mastering for single axis executes the quick mastering for the selected axes. In the case where it is difficult to move all axes to their reference positions, user can execute the quick mastering to the selected axis, only if they are at the reference positions.

```
QUICK MASTER FOR SINGLE AXIS
                                          1/9
         ACTUAL POS  ( REF POS)   (SEL)[ST]
       J1   25.225 (   0.000)    (0)  [2]
       J2   25.550 (   0.000)    (0)  [2]
       J3  -50.000 (   0.000)    (0)  [2]
       J4   12.500 (   0.000)    (0)  [2]
       J5   31.250 (   0.000)    (0)  [0]
       J6   43.382 (   0.000)    (0)  [0]
       E1    0.000 (   0.000)    (0)  [0]
       E2    0.000 (   0.000)    (0)  [0]
       E3    0.000 (   0.000)    (0)  [0]

                                  EXEC
```

Table B.2.4 Setting for “Quick mastering for single axis”

Items	Descriptions
ACTUAL POS	The current position expressed by joint angle (degree) of the robot.
REF POS	The reference position expressed by joint angle (degree) of the robot.
SEL	Set 1 to this item, of the axis to perform mastering for. Other axes should be 0.

Items	Descriptions
ST	Display the completion status of the single axis mastering. The value displayed at this item cannot be directly changed. - 0 Specifies that the mastering data has been lost. The "Quick mastering for single axis" needs to be performed. - 2 The mastering has been completed.

For example, when J3 mastering data is lost, user can select the J3 axis and execute the quick mastering only for J3 axis. As illustrated in Fig. B.2.4.1, the J3 joint angle should match that of reference position at the mastering.

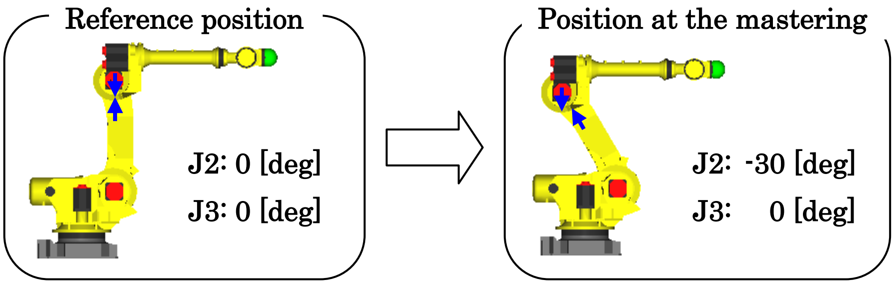


Fig. B.2.4(a) Example of position at the quick mastering for J3 axis

As in the quick mastering, when you move the selected axis to the reference position, the misalignment from the reference position should be within half revolution of the motor.

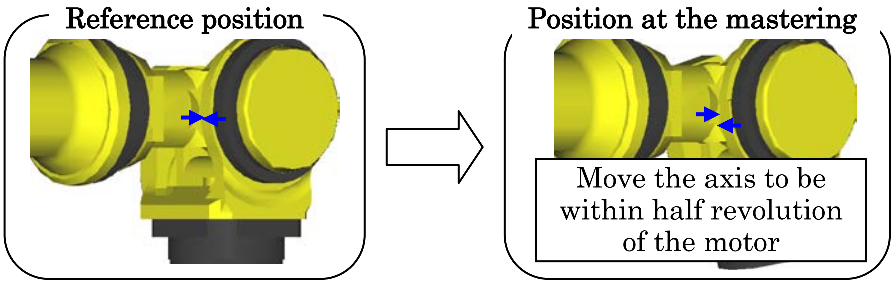


Fig. B.2.4(b) Example of position at the quick mastering for J5 axis

The reference position of the "Quick mastering for single axis" is same position with the quick mastering. Please refer to B.2.3 "Quick Mastering" for the detail.

Procedure B-10 Quick mastering for single axis

Condition

1. System variable $MASTER_ENB must be set to 1.
2. Quick mastering reference position (REF POS) must be set.
3. For the robot model that includes Gravity Compensation (robot arm weight only) by default, tentative mastering has been performed for all the robot axes. (Refer to "Mastering when Gravity Compensation is enabled" at B.2 MASTERING.)

Step

1. Press [MENU] key. The screen menu will be displayed. Select " 0 — NEXT —" and then select " 6 SYSTEM" . Press F1, [TYPE]. The screen change menu will be displayed.
2. Select "Master/Cal" on the screen change menu. The Master/Cal screen will be displayed.
3. Select "4 QUICK MASTER FOR SINGLE AXIS". The "Quick mastering for single axis" screen is displayed.

```
SYSTEM Master/Cal
                                       TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE

      Press 'ENTER' or number key to select.

   [ TYPE        RES_PCA          DONE
```

```
QUICK MASTER FOR SINGLE AXIS
                                              1/9
         ACTUAL POS  ( REF POS)   (SEL)[ST]
       J1   25.225 (   0.000)    (0)  [2]
       J2   25.550 (   0.000)    (0)  [2]
       J3  -50.000 (   0.000)    (0)  [2]
       J4   12.500 (   0.000)    (0)  [2]
       J5   31.250 (   0.000)    (0)  [0]
       J6   43.382 (   0.000)    (0)  [0]
       E1    0.000 (   0.000)    (0)  [0]
       E2    0.000 (   0.000)    (0)  [0]
       E3    0.000 (   0.000)    (0)  [0]

                                  EXEC
```

In this example, the mastering of J5 and J6 axis needs to be executed.

4 Enter 1 to SEL setting field of the axis that you want to perform the mastering. SEL can be specified one axis at a time or plural axes simultaneously.

```
QUICK MASTER FOR SINGLE AXIS

       J5   31.250 (   0.000)    (1)  [0]
       J6   43.382 (   0.000)    (1)  [0]

                                  EXEC
```

5 Jog the robot to the mastering position. Turn off the brake control if it is necessary.
6 Press F5, EXEC. The mastering is performed. This operation sets 0 to SEL and 2 to ST.

```
QUICK MASTER FOR SINGLE AXIS
                                              1/9
         ACTUAL POS  ( REF POS)   (SEL)[ST]
       J1   25.225 (   0.000)    (0)  [2]
       J2   25.550 (   0.000)    (0)  [2]
       J3  -50.000 (   0.000)    (0)  [2]
       J4   12.500 (   0.000)    (0)  [2]
       J5    0.011 (   0.000)    (0)  [2]
       J6    0.024 (   0.000)    (0)  [2]
       E1    0.000 (   0.000)    (0)  [0]
       E2    0.000 (   0.000)    (0)  [0]
       E3    0.000 (   0.000)    (0)  [0]

                                  EXEC
```

7 When the single axis mastering is completed, press the [PREV] key to display the Master/Cal screen.

```
SYSTEM Master/Cal
                                          TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE

      Press 'ENTER' or number key to select.

  [ TYPE ]         RES_PCA           DONE
```

8 Select “7 CALIBRATE” and press F4, YES. The calibration is performed.
9 Press F5, DONE, after calibration.

> ⚠ **CAUTION**
> After the quick mastering, set reference position for quick mastering by executing “6 SET QUICK MASTER REF”, providing for future quick mastering.

B.2.5 Single Axis Mastering

The 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 the Pulsecoder or exchanging of the Pulsecoder.

```
SINGLE AXIS MASTER
                                              1/9
       ACTUAL POS   (MSTR POS )  (SEL)[ST]
     J1    25.225   (    0.000)  (0)  [2]
     J2    25.550   (    0.000)  (0)  [2]
     J3   -50.000   (    0.000)  (0)  [2]
     J4    12.500   (    0.000)  (0)  [2]
     J5    31.250   (    0.000)  (0)  [2]
     J6    43.382   (    0.000)  (0)  [2]
     E1     0.000   (    0.000)  (0)  [0]
     E2     0.000   (    0.000)  (0)  [0]
     E3     0.000   (    0.000)  (0)  [0]

                             GROUP    EXEC
```

Table B.2.5(a) 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 cannot be directly changed. - 0 The mastering data has been lost. The single axis mastering needs to be performed. - 1 The mastering data has been lost (the mastering is done only on the other interaction axis). The single axis mastering needs to be performed. For example, when J2/J3 axis is an interaction pair, J3-axis ST becomes 1 after you perform single axis mastering on J2 axis. - 2 The mastering has been completed.

Refer to the mechanical manual for an accurate method of single axis mastering.

Single axis mastering for interaction axis

When single axis mastering is done in interaction axis, the axis of the interaction pair is also influenced. Therefore, mastering of these interaction axes must be done at the same time. Interaction axis depends on the robot model. Following table shows the relation between robot model and interaction axis. For example, when robot model is R-2000*i*B/165F and J4 motor is changed, mastering of J5 and J6 should be done at the same time with J4. (Provided that if 7DC2(V8.20P)/14 or later software version is installed, axis numbers which should be selected to do single axis mastering at the same time are displayed if these are not selected. In this case, it is unnecessary to refer to a following table.)

Table B.2.5(b) Relation between robot model and interaction axis

Robot model		Interaction axis
R-2000*i*B	/100P, /125L, /150U, /165CF, /165EW, /165F, /165R, /170CF, /175L, /185L, /200EW, /200R, /200T, /210F, /210F/STUD, /210WE, /220U, /250F	・ J2/J3 ・ J4/J5/J6
	/100H, /100H-2	・ J2/J3 ・ J4/J5
	/210FS, /220US	・ J2/J3 ・ J5/J6
R-2000*i*C	/165F, /210F	・ J2/J3 ・ J4/J5/J6
R-1000*i*A	/80F, /100F	・ J2/J3 ・ J4/J5/J6
	/80H	・ J2/J3/J4/J5
M-2000*i*A	/900L, /1200	・ J4/J5/J6
M-900*i*A	/150P, /200P, /260L, /350, /400L, /600	・ J4/J5/J6
M-900*i*B	/360, /400L, /700	・ J4/J5/J6
M-710*i*C	/50, /50E, /50S, /50T, /50T-HS, /70, /70T, /70T(Alpha-iS30/3000), /70W Long-Left, /70W Long-Right, /70W Short-Left, /70W Short-Right	・ J2/J3 ・ J4/J5/J6
	/20L	・ J2/J3
	/50H	・ J2/J3 ・ J4/J5
M-410*i*B	/140H	・ J3/J4/J5
	/160, /300, /450, /700	Nothing
M-410*i*C	/185, /315	Nothing
M-420*i*A	M-420*i*A, High Speed Wrist, Second Food	Nothing
M-421*i*A	M-421*i*A	Nothing
M-430*i*A	/2F, /2FH, /4FH	・ J1/J2/J3 ・ J4/J5
	/2P, /2P(S), /2PH	・ J1/J2/J3 ・ J4/J5/J6
M-6*i*B ARC Mate 100*i*B ROBOWELD 100*i*B	M-6*i*B (ARC Mate 100*i*B, ROBOWELD 100*i*B), /2HS, /2HS MH SA, /6C, /6S, /6T, SA, e SA, e	・ J2/J3
M-16*i*B ARC Mate 120*i*B ROBOWELD 120*i*B	M-16*i*B (ARC Mate 120*i*B, ROBOWELD 120*i*B), /10L, /10LT, /10L Solution Arm, /20, /20T, e, Solution Arm, e/FD DP	・ J2/J3
M-10*i*A ARC Mate 100*i*C ROBOWELD 100*i*C	M-10*i*A (ARC Mate 100*i*C, ROBOWELD 100*i*C), /6L, /10S, /12, e, e/6L	・ J2/J3 ・ J5/J6
	/10M, /10MS	・ J2/J3
M-20*i*A ARC Mate 120*i*C ROBOWELD 120*i*C	/20M, /35M	・ J2/J3
	M-20*i*A (ARC Mate 120*i*C, ROBOWELD 120*i*C), /10L, /20T	・ J2/J3 ・ J5/J6
ARC Mate 0*i*A R-0*i*A	ARC Mate 0*i*A (R-0*i*A)	・ J2/J3

Robot model		Interaction axis
ARC Mate 0iB R-0iB	ARC Mate 0iB (R-0iB)	・ J2/J3
M-1iA	/0.5A, /0.5AL	・ J4/J5/J6
	/0.5S, /0.5SL, /1H, /1HL	Nothing
M-2iA	/3S, /3SL, /6H, /6HL	Nothing
M-3iA	/6A	・ J4/J5/J6
	/6S, /6S(B203), /12H	Nothing
LR Mate 200iC ARC Mate 50iC ROBOWELD mini C	LR Mate 200iC (ARC Mate 50iC, ROBOWELD mini C), /5L, /5LC, /5WP	・ J2/J3 ・ J5/J6
	/5C, /5F, /5H, /5H High Speed	・ J2/J3 ・ J4/J5
LR Mate 200iD ARC Mate 50iD	LR Mate 200iD (ARC Mate 50iD), /4S, /4SC, /4SH, /7C, /7H, /7L, /7LC, /7WP	・ J2/J3 ・ J5/J6
	/4SH, /7H	・ J2/J3 ・ J4/J5
	4SH-2, 7H-2	・ J2/J3/J4/J5

NOTE

After mastering, please check whether mastering has been done correctly by using zero position mark (witness mark).
After checking mastering has been done correctly, please confirm the behavior of the robot and the robot system.

Procedure B-11 Single axis mastering

Condition

1. System variable $MASTER_ENB must be set to 1.
2. For the robot model that includes Gravity Compensation (robot arm weight only) by default, tentative mastering has been performed for all the robot axes. (Refer to “Mastering when Gravity Compensation is enabled” at B.2 MASTERING.)

Step

1. Press [MENU] key. The screen menu will be displayed.
2. Select ”0 ー NEXT ー” and then select ”6 SYSTEM”.
3. Press F1, [TYPE]. The screen change menu will be displayed.
4. Select “Master/Cal” on the screen change menu. The Master/Cal screen will be displayed.

```
SYSTEM Master/Cal
                              TORQUE = [ON  ]
    1 FIXTURE POSITION MASTER
    2 ZERO POSITION MASTER
    3 QUICK MASTER
    4 QUICK MASTER FOR SINGLE AXIS
    5 SINGLE AXIS MASTER
    6 SET QUICK MASTER REF
    7 CALIBRATE

     Press 'ENTER' or number key to select.

   [ TYPE ]           RES_PCA            DONE
```

5. Select ”5 SINGLE AXIS MASTER”. The single axis mastering screen is displayed.

```
SYSTEM Master/Cal
                             TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE

      Press 'ENTER' or number key to select.

  [ TYPE ]            RES_PCA             DONE
```

```
SINGLE AXIS MASTER
                                          1/9
       ACTUAL POS   (MSTR POS )  (SEL)[ST]
     J1    25.225   (    0.000)  (0)  [2]
     J2    25.550   (    0.000)  (0)  [2]
     J3   -50.000   (    0.000)  (0)  [2]
     J4    12.500   (    0.000)  (0)  [2]
     J5    31.250   (    0.000)  (0)  [0]
     J6    43.382   (    0.000)  (0)  [0]
     E1     0.000   (    0.000)  (0)  [0]
     E2     0.000   (    0.000)  (0)  [0]
     E3     0.000   (    0.000)  (0)  [0]

                                GROUP     EXEC
```

In this example, the mastering of J5 and J6 axis needs to be executed.

6 Enter 1 to SEL setting field of the axis that you want to perform the mastering. SEL can be specified one axis at a time or plural axes simultaneously.

```
SINGLE AXIS MASTER

     J5    31.250   (    0.000)  (1)  [0]
     J6    43.382   (    0.000)  (1)  [0]

                                GROUP     EXEC
```

7 Jog the robot to the mastering position. Turn off the brake control if it is necessary.
8 Enter the axis data of the mastering position.

```
SINGLE AXIS MASTER

     J5    31.250   (    0.000)  (1)  [0]
     J6    43.382   (   90.000)  (1)  [0]

                                GROUP     EXEC
```

9 Press F5, EXEC. The mastering is performed. This operation sets 0 to SEL and 1 or 2 to ST.

```
SINGLE AXIS MASTER
                                                  1/9
       ACTUAL POS   (MSTR POS )  (SEL)[ST]
     J1    25.225   (     0.000)  (0)  [2]
     J2    25.550   (     0.000)  (0)  [2]
     J3   -50.000   (     0.000)  (0)  [2]
     J4    12.500   (     0.000)  (0)  [2]
     J5     0.000   (     0.000)  (0)  [2]
     J6    90.000   (    90.000)  (0)  [2]
     E1     0.000   (     0.000)  (0)  [0]
     E2     0.000   (     0.000)  (0)  [0]
     E3     0.000   (     0.000)  (0)  [0]

                              GROUP    EXEC
```

10 When the single axis mastering is completed, press the [PREV] key to display the Master/Cal screen.

```
SYSTEM Master/Cal
                              TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 QUICK MASTER FOR SINGLE AXIS
     5 SINGLE AXIS MASTER
     6 SET QUICK MASTER REF
     7 CALIBRATE

      Press 'ENTER' or number key to select.

   [ TYPE ]          RES_PCA           DONE
```

11 Select ”7 CALIBRATE” and press F4, YES. The calibration is performed.
12 Press F5, DONE, after calibration.

B.2.6 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-12 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 will be displayed.

```
SYSTEM Variables
                                        1/638
      1 $AAVM            AAVM_T
      2 $ABSPOS_GRP      ABSPOS_GRP_T
      3 $ACC_MAXLMT      150
      4 $ACC_MINLMT      0
      5 $ACC_PRE_EXE     0
      6 $ALM_IF          ALM_IF_T
      7 $ANGTOL          [9] of REAL
      8 $APPLICATION     [9] of STRING[21]
      9 $AP_ACTIVE       6
     10 $AP_AUTOMODE     FALSE
     11 $AP_CHGAPONL     TRUE

  [ TYPE ] DETAIL
```

3 Change mastering data.
Mastering data is stored in system variable $DMR_GRP.$MASTER_COUN.

```
SYSTEM Variables

    114 $DMR_GRP         DMR_GRP_T
    115 $DMSW_CFG        DMSW_CFG_T

  [ TYPE ] DETAIL
```

4 Select $DMR_GRP.

```
SYSTEM Variables
    $DMR_GRP                            1/1
      1   [1]            DMR_GRP_T

  [ TYPE ] DETAIL
```

```
SYSTEM Variables
    $DMR_GRP[1]                         4/29
      1 $MASTER_DONE     FALSE
      2 $OT_MINUS        [9] of BOOLEAN
      3 $OT_PLUS         [9] of BOOLEAN
      4 $MASTER_COUN     [9] of INTEGER
      5 $REF_DONE        FALSE
      6 $REF_POS         [9] of REAL
      7 $REF_COUNT       [9] of INTEGER
      8 $BCKLSH_SIGN     [9] of BOOLEAN
      9 $EACHMST_DON     [9] of INTEGER
     10 $SPC_COUNT       [9] of INTEGER
     11 $SPC_MOVE        [9] of BOOLEAN

  [ TYPE ]                  TRUE     FALSE
```

5 Select “$MASTER_COUN” and enter mastering data.

```
SYSTEM Variables
    $DMR_GRP[1].$MASTER_COUN          1/9
      1   [1]             95678329
      2   [2]             10223045
      3   [3]             3020442
      4   [4]             304055030
      5   [5]             20497709
      6   [6]             2039490
      7   [7]             0
      8   [8]             0
      9   [9]             0

   [ TYPE ]
```

6 Press the [PREV] key.
7 Set "$GRAV_MAST". If you have recorded (written down) the value of this system variable together with $MASTER_COUN, enter the value (0 or 1). If not, enter -1.

```
SYSTEM Variables
    $DMR_GRP[1]                        29/29
     29 $GRAV_MAST        -1

   [ TYPE ]
```

8 Set "$MASTER_DONE" to "TRUE".

```
SYSTEM Variables
    $DMR_GRP[1]                        1/29
      1 $MASTER_DONE      TRUE
      2 $OT_MINUS         [9] of BOOLEAN

   [ TYPE ]                      TRUE      FALSE
```

9 Display the Master/Cal screen and select "7 CALIBRATE".
10 Press F5, DONE, after calibration.

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  CONFIG    MOTOR     SERVO
```

- F2, SOFTWARE : Displays the software version screen.
- F3, CONFIG : Displays the software configuration screen.
- F4, MOTOR : Displays the motor information screen.
- F5, SERVO : Displays the servo parameter information screen.

Software version screen

The software version screen displays the following information:

```
STATUS Version ID
        SOFTWARE:              ID:    1/19
    1  HandlingTool            7DC1/01
    2  S/W Serial No.          88150
    3  Controller ID           F00000
    4  Default Personality (from FD)
    5  R-2000iB/210F           V8.10P/01
    6  Servo Code              12345678
    7  Cart. Mot. Parameter    V3.00
    8  Joint Mot. Parameter    V3.00
    9  DCS                     V3.0.0
   10  Stop pattern            A
   11  Version ID              V8.10P/01

   [ TYPE ] SOFTWARE  CONFIG   MOTOR   SERVO
```

Software configuration

The software configuration screen displays the software installed.

```
STATUS Version ID
        FEATURE:             ORD NO:  1/333
    1  HandlingTool           H552
    2  Japanese (KANA) Dict   H520
    3  Multi Language (ENGL)  H531
    4  AA Vision Mastering    AAVM
    5  Analog I/O             H550
    6  Auto Software Update   AUTP
    7  Automatic Backup       J545
    8  Background Editing     J616
    9  Camera I/F             VCAM
   10  Cell I/O               CLIO
   11  Common shell           R645

   [ TYPE ] SOFTWARE  CONFIG   MOTOR   SERVO
```

Motor information screen

The motor information screen displays servo control information on each axis.

```
STATUS Version ID
      GR: AX: MOTOR ID AND INFO:     1/56
    1  1   1  ACaiSR30/3000 80A H1 DSP1-L
    2  1   2  ACaiSR30/3000 80A H2 DSP1-M
    3  1   3  ACaiSR30/3000 80A H3 DSP1-J
    4  1   4  Aca12/4000iS 40A  H4 DSP1-K
    5  1   5  Aca12/4000iS 40A  H5 DSP2-L
    6  1   6  Aca12/4000iS 40A  H6 DSP2-M
    7 **  **  ****************************
    8 **  **  ****************************
    9 **  **  ****************************
   10 **  **  ****************************
   11 **  **  ****************************

   [ 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
        GROUP: AXIS:  SERVO PARAM ID   1/56
    1   1      1     P02.06
    2   1      2     P02.06
    3   1      3     P02.06
    4   1      4     P02.06
    5   1      5     P02.06
    6   1      6     P02.06
    7  **     **     ********
    8  **     **     ********
    9  **     **     ********
   10  **     **     ********
   11  **     **     ********

   [ TYPE ] SOFTWARE  CONFIG  MOTOR  SERVO
```

Procedure B-12   Software version screen

Step

1. Press [MENU] 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
        SOFTWARE:              ID:    1/19
    1   HandlingTool            7DC1/01
    2   S/W Serial No.          88150
    3   Controller ID           F00000
    4   Default Personality (from FD)
    5   R-2000iB/210F           V8.10P/01
    6   Servo Code              12345678
    7   Cart. Mot. Parameter    V3.00
    8   Joint Mot. Parameter    V3.00
    9   DCS                     V3.0.0
   10   Stop pattern            A
   11   Version ID              V8.10P/01

   [ 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 display 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

                                    GRP[ 1 ]
          Flag Bits 1/2          History
     J1:  0000000000000000 (0000000000000000)
          0000000000000000 (0000000000000000)
     J2:  0000000000000000 (0000000000000000)
          0000000000000000 (0000000000000000)
     J3:  0000000000000000 (0000000000000000)
          0000000000000000 (0000000000000000)
     J4:  0000000000000000 (0000000000000000)
          0000000000000000 (0000000000000000)

   [TYPE]  STATUS1  STATUS2  PULSE  [UTIL ]  >
```

Flag 1	Servo alarm status 1
Flag 2	Servo alarm status 2

Table B.4 (a) Servo alarm status 1

MSB	B14	B13	B12	B11	B10	B9	B8
		OVC				FBAL	ALDF
B7	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**LSB**
	MOFAL	EROFL	CUER			SRDY	

OVC	It indicates an overcurrent (OVC) alarm.
FBAL	Disconnection alarm (ALDF indicates whether the disconnection is associated with the hardware or software.)
ALDF	Alarm distinction bit
MOFAL	Motion command overflow alarm
EROFL	Error counter overflow alarm for line tracking
CUER	Current offset error
SRDY	Servo ready signal

CLALM	FBAL	ALDF	Alarm
0	1	0	Pulsecoder disconnection alarm detected by the software
0	1	1	Pulsecoder disconnection alarm detected by the hardware (not used for serial Pulsecoder)
1	0	0	Collision detection alarm
1	0	1	Tip stick detection

Table B.4 (b) Servo alarm status 2

MSB	B14	B13	B12	B11	B10	B9	B8
SRCMF	CLALM						
B7	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**LSB**
FSSBDC	SVUCAL	AMUCAL		NOAMP			

SRCMF	Compensation warning flag When part of the position data is missing because of noise or some other reason, this bit indicates that data compensation is performed.
CLALM	It indicates a collision detection alarm.
FSSBDC	FSSB disconnection alarm (Hardware detection by FSSBDC)
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)
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.

⚠ CAUTION
The bits which are not described in above table are used to monitor the various statuses for control.

Status 2 screen

The status 2 screen indicates the Pulsecoder alarm status (12 bits).

```
STATUS Axis

                                    GRP[ 1 ]
          Alarm Status     History
     J1 : 000000000000 ( 000000000000 )
     J2 : 000000000000 ( 000000000000 )
     J3 : 000000000000 ( 000000000000 )
     J4 : 000000000000 ( 000000000000 )
     J5 : 000000000000 ( 000000000000 )
     J6 : 000000000000 ( 000000000000 )

   [ TYPE ] STATUS1 STATUS2  PULSE  [UTIL ]  >
```

Alarm Status	Pulsecoder alarm status

Table B.4 (c) Pulsecoder alarm status

				MSB	**B10**	**B9**	**B8**
				SPHAL	STBERR	CRCERR	DTERR
B7	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**LSB**
OHAL	LDAL	BLAL	PHAL	CMAL	BZAL	PMAL	

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 an over heat alarm.
LDAL	When this bit is 1, it indicates a diode abnormal 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.
CMAL	When this bit is 1, it indicates a count miss alarm.
BZAL	When this bit is 1, it indicates an exhausted battery alarm.
PMAL	When this bit is 1, it indicates a pulse miss alarm.

Pulse screen

The pulse screen displays the servo delay, machine position, and status of the motion command.

```
STATUS Axis

                                   GRP[ 1 ]
          Position    Machine     Motion
            Error       Pulse       Command
     J1 :         0           0           0
     J2 :         0           0           0
     J3 :         0           0           0
     J4 :         0           0           0
     J5 :         0           0           0
     J6 :         0           0           0

   [ 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, over travel, and servo amplifier. Load to the motor and thermal loss can be estimated using the root-mean-square current values.

```
STATUS Axis

                                   GRP[ 1 ]
         Torque Monitor
               Ave. / Max.    Inpos OT VRDY
     J1 :      0.000/     0.000  1   0  OFF
     J2 :      0.000/     0.000  1   0  OFF
     J3 :      0.000/     0.000  1   0  OFF
     J4 :      0.000/     0.000  1   0  OFF
     J5 :      0.000/     0.000  1   0  OFF
     J6 :      0.000/     0.000  1   0  OFF

   [ 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	Over travel 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

      Tracking Status

           Flag Bits 1      Flag Bits 2
     P1 : 0000000000000000 0000000000000000
     P2 : 1111111111111111 0111111111111111

          Alarm Status    Counter Value
     P1 : 000000000000            0
     P2 : 111111111111        ********

   [ TYPE ] MONITOR TRACKING DISTURB          >
```

Flag Bits 1	Servo alarm status 1
Flag Bits 2	Servo alarm status 2
Alarm Status	Pulsecoder alarm status
Counter Value	Line tracking counter

For the servo and Pulsecoder alarm statuses, see Table B.4 (a), Table B.4 (b), and Table B.4 (c).

Disturbance torque screen

The disturbance torque screen displays the disturbance torque to each motor (current torque and maximum and minimum torque for each ITP). The disturbance torque is indicated with the current values estimated from the difference between the scheduled and actual values of the Pulsecoder. If the maximum or minimum value set for the disturbance torque is exceeded, the collision detection function of the servo system regards a collision as occurring and turns the servo power off.

The estimated disturbance torque value is changed in real time. The “Max.” and “Min.” indicate the maximum and minimum value of the estimated disturbance torque value from the start of the program execution or the start of jog feed.
“Allowed” is the value that is allowed a certain amount of the torque for the threshold value to detect the collision. In order to prevent detecting by mistake, adjust the program so that the maximum and minimum value of the disturbance torque are in range specified in “Allowed”.

```
STATUS Axis

                                  GRP[ 1 ]
         Disturbance Torque (A)
         Curr./ Max.(Allowed)/Min.(Allowed)
    J1 :   0.0    0.0(  90.9)    0.0( -90.9)
    J2 :   0.0    0.0(  84.3)    0.0( -84.3)
    J3 :   0.0    0.0(  97.4)    0.0( -97.4)
    J4 :   0.0    0.0(  30.2)    0.0( -30.2)
    J5 :   0.0    0.0(  34.3)    0.0( -34.3)
    J6 :   0.0    0.0(  21.7)    0.0( -21.7)

   [ TYPE ] 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 [MENU] key to display the screen menu.
2 Select “0 ─ NEXT ─” to display the next page, then select “4 STATUS”.
3 Press the F1, [TYPE] key to display the screen change menu.
4 Select “Axis.” The robot axis status screens can be displayed.

```
STATUS Axis

                                  GRP[ 1 ]
           Flag Bits 1/2         History
     J1:  0000000000000000 (0000000000000000)
          0000000000000000 (0000000000000000)
     J2:  0000000000000000 (0000000000000000)
          0000000000000000 (0000000000000000)
     J3:  0000000000000000 (0000000000000000)
          0000000000000000 (0000000000000000)
     J4:  0000000000000000 (0000000000000000)
          0000000000000000 (0000000000000000)

 [ TYPE ] STATUS1 STATUS2  PULSE   [UTIL ]  >
 [ TYPE ] MONITOR TRACKING DISTURB [UTIL ]  >
 [ TYPE ]                  DIAG.   [UTIL ]  >
```

- F2, STATUS 1 : Displays the status 1 screen.
- F3, STATUS 2 : Displays the status 2 screen.
- F4, PULSE : Displays the pulse screen.
- F2, MONITOR on the next page : Displays the monitor screen.
- F3, TRACKING on the next page : Displays the tracking screen.
- F4, DISTURB on the next page : Displays the disturbance torque screen.
- When F4, DIAG. is selected on the next page, the diagnosis screen will be displayed.

5 To change the group number, press F5, [UTIL]. A menu will be displayed. 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
- Torque diagnosis
- Disturbance diagnosis
- OVC diagnosis
- Collision diagnosis
- Help

B.5.2 About Reducer Diagnosis

When the exchange or overhaul of the reducer is done, you have to reset the value to calculate the lifetime of reducer accurately.
The procedure to reset the calculated lifetime value of reducer is as follows.

- Please push E-Stop. The calculation of the lifetime of reducer pauses.
- Please set $FMS_GRP[reducer exchanged group].$T_LIFE[reducer exchanged axis] to 0 in the system variable screen.
  The data to calculate the lifetime of reducer are reset.

> ⚠ **CAUTION**
> Do not set the variable of the other axes .

- Please reset E-stop. The calculation of the lifetime of reducer restarts.

B.5.3 Procedure

Procedure B-14 Diagnosis screen

Step

1. Press [MENU] 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 the [NEXT] key until “DIAG.” is shown above function key.
6. Press F4, DIAG. key. Diagnosis main screen is shown first. The currently selected jog group will be displayed.

Limitations

- When the group 1 does not support the screen, it will not be displayed.
- When the currently selected jog group does not support the screen, the group 1 will be displayed.

Procedure B-15 Change diagnosis screen

Step

1. Each item is allocated to the function key. Press function key to show the item.
2. You can change the allocation of function keys by pressing the [NEXT] key.
3. To show the Axis screen again, press the [PREV] key.

Switching group

To switch the group to be displayed, switch the jog group, then press the function key of the selected item again.

B.5.4 Each Item

Main: Each item shows the value of the worst axis.

```
Diagnosis
                                        1/6
     group  [1]
     reducer                 50.00 %
     current                 70.00 %
     disturbance             30.00 %
     OVC                     20.00 %
     collision detection        10 times
     discharge                 100 W

   [ TYPE ]   main   reducer          HELP   >
```

Reducer: The ratio how much the reducer achieves to the goal of lifetime (L10 lifetime).

```
Diagnosis/reducer
                                        1/1
     group [1]       used(estimated)
            J1     50.0%
            J2     17.0%
            J3     18.0%
            J4      5.0%
            J5     40.0%
            J6      3.0%
     100%(estimated) means L10 life achieved.

   [ TYPE ]   main   reducer          HELP   >
```

Torque: The ratio of the current torque to the maximum.

```
Diagnosis
                                        1/1
     torque
     group [1]
          J1    46.21 %
          J2    63.58 %
          J3    50.13 %
          J4    30.50 %
          J5    15.09 %
          J6    56.97 %

   [ TYPE ]   main   reducer          HELP   >
```

Disturbance: The ratio of the force observed by the servo software to the alarm threshold.

```
Diagnosis
                                          1/1
    disturbance
    group [1] current  max(%)    min(%)
         J1   16.21 %  20.55 /   -12.34
         J2   23.58 %  30.77 /    -0.12
         J3  -10.13 %  10.66 /   -34.56
         J4   20.50 %  30.44 /   -27.89
         J5   15.09 %  20.23 /   -17.65
         J6   26.97 %  26.97 /   -21.34

   [ TYPE ]   main   reducer            HELP   >
```

OVC: The ratio of the motor temperature simulated by the software to the alarm threshold.

```
Diagnosis
                                          1/1
    OVC
    group [1]
         J1    16.21 %
         J2    13.58 %
         J3    10.13 %
         J4    20.50 %
         J5    15.09 %
         J6    26.97 %

   [ TYPE ]   main   reducer            HELP   >
```

Collision detection: The count of the collision and the data of the last collision detection.

```
Diagnosis
                                          1/2
    last detection
      2010/  4/  5, 13: 59: 18
    group [1]
    count / position
    J1  *****  times   12.23 deg
    J2  *****  times  -22.23 deg
    J3  *****  times   32.23 deg
    J4  *****  times –126.23 deg
    J5     24  times -112.23 deg
    J6  *****  times  152.23 deg

   [ TYPE ]   main   reducer   ov.heat   HELP   >
```

Help: Information about the meaning and the remedy of the last shown item

```
Diagnosis
    HELP      Arrows to scroll, PREV to exit
     INFORMATION:
     Each item shows the value of the worst
     axis.
     REMEDY:


     DETAIL:
     You can also watch the data of all
     axes by pushing the function keys.
     Please  utilize [prev],[next] key
     too.
```

B.6 WORLD FRAME ORIGIN

This section describes the world frame origin of the each robot model (See Section 3.9,"SETTING COORDINATE SYSTEMS" for the world frame). When the user frame or tool frame is set, refer to this.

R series / M series/ ARC Mate / LR Mate

(Other than Top Mount Robot)
A crossing point between J1 axis and level plane which includes the J2 axis.
(Top Mount Robot)
The position on J1 axis that is closest to J4 axis when J1 axis is at the zero position.

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 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.

For connection of the I/O Link, refer to the "FANUC Robot series R-30*i*B/R-30*i*B Plus CONTROLLER MAINTENANCE MANUAL" (B-83195EN) or the "FANUC Robot series R-30*i*B Mate/R-30*i*B Mate Plus CONTROLLER MAINTENANCE MANUAL" (B-83525EN).

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 DI[i]/DO[i]
- Group I/O GI[i]/GO[i]
- Analog I/O AI[i]/AO[i]
- Peripheral 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.7 (a) I/O modules

	Abbreviation
Process I/O board	-
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	Connection unit

I/O assignment

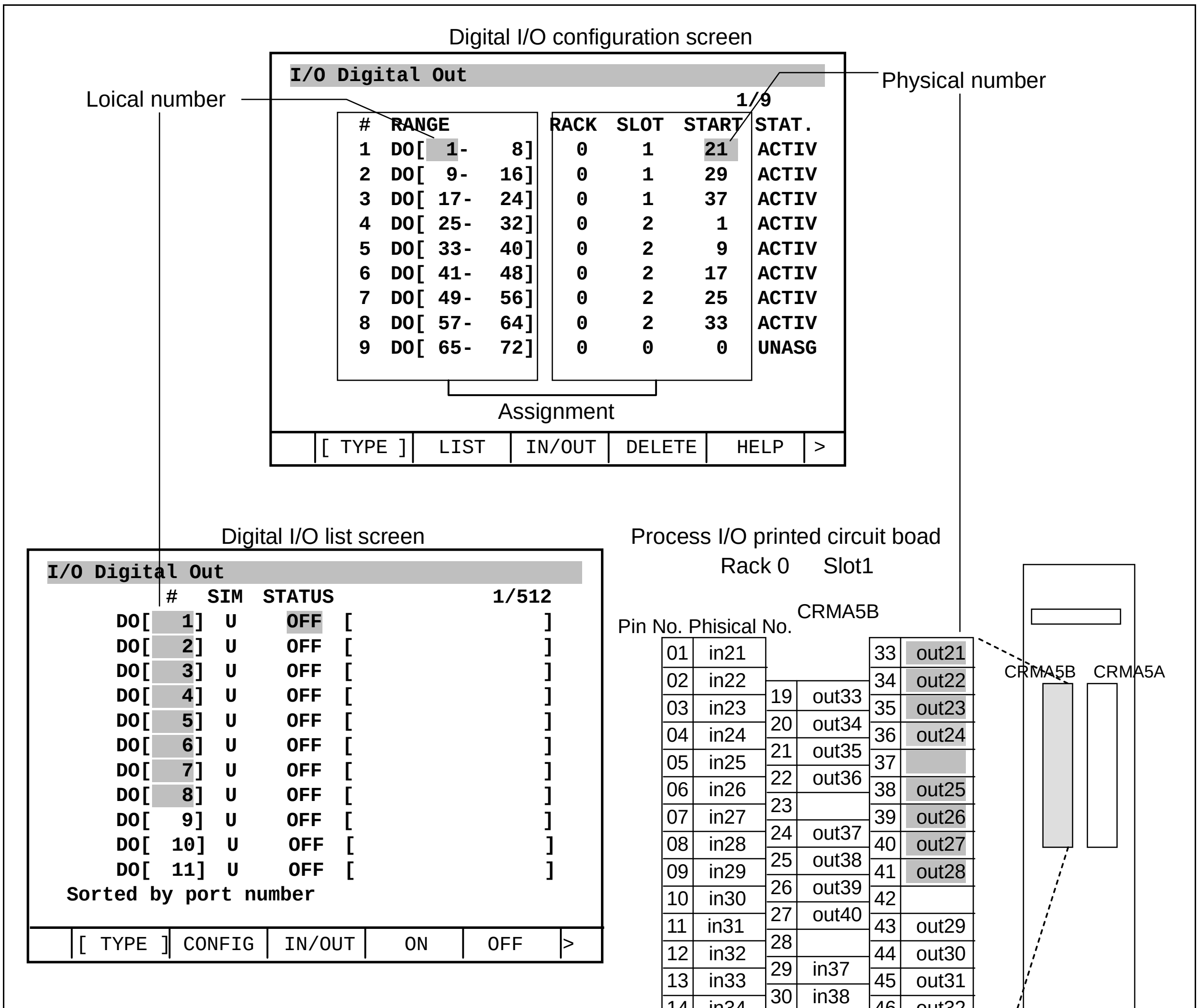
I/O logical number i is assigned to a physical number of I/O modules. I/O logical numbers can be redefined. Refer to "3.1 I/O" for detail of I/O assignment.

- 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.



On the configuration screen, DO[1] to DO[8] are assigned to out21 to out28 of connector CRMA5B on the process I/O printed circuit board. When DO[1] is set to ON, a signal is output on pin 33.

Fig. B.7 (a) Logical number and physical number assignment

RACK

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

Slot numbers indicate the module number which the I/O module is composed. 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 (CHANNEL)

Digital I/Os and peripheral I/Os are assigned in groups of any signals. Specify the first physical number for the 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 (b) Specifying rack and slot numbers for each I/O module

I/O module	Rack	Slot
Process I/O printed circuit board	Always 0	(NOTE 2)
FANUC I/O Unit-MODEL A	(NOTE 1)	Number indicated on the base unit
FANUC I/O Unit-MODEL B	(NOTE 1)	Unit number (set with DIP switches)
FANUC I/O Link connection unit	Always 0	(NOTE 2)

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.
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.

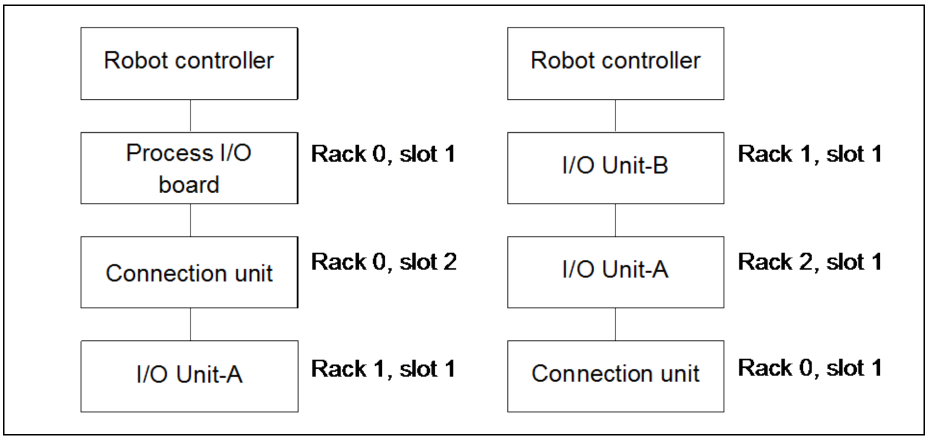


Fig. B.7 (b) Example of rack and slot specification

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, cycle power of the controller. Data assignment is performed automatically.

- When additional specification is necessary

Specify the number of input and output signals in I/O link screen in the Robot controller.

	Specification
Process I/O board	Unnecessary
FANUC I/O Unit-MODEL A	Unnecessary
FANUC I/O Unit-MODEL B	Necessary
FANUC I/O Link connection unit	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.

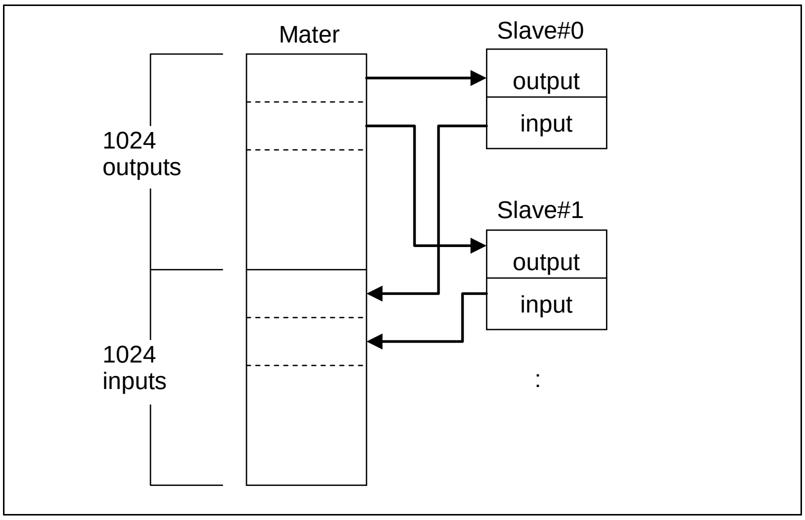


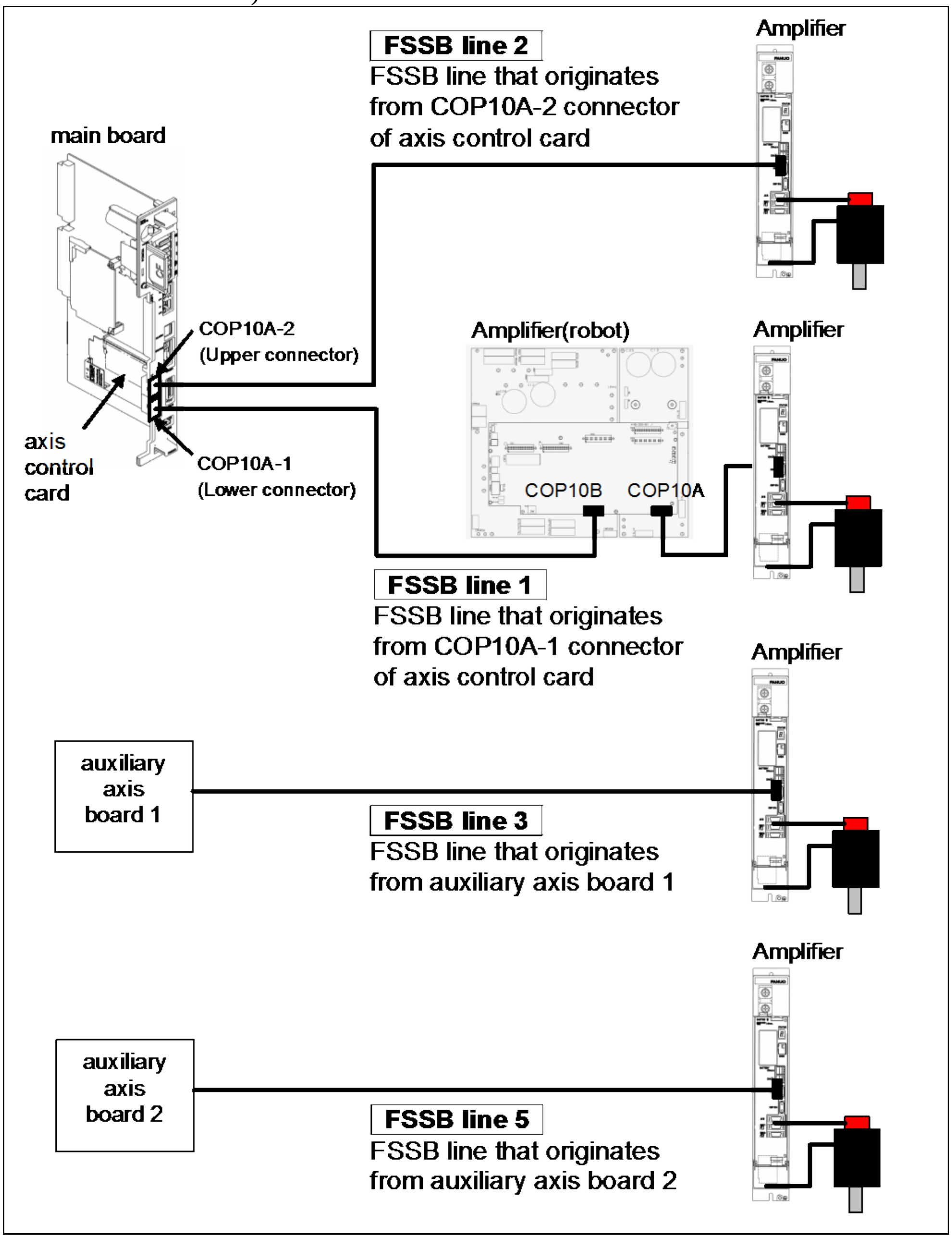
Fig. B.7 (c) Relation between master and slave in I/O signal points

B.8 FSSB LINE SETUP

This document describes the FSSB LINE and HARDWARE START AXIS setting that is necessary to set up when system has auxiliary axes.

B.8.1 Definition of FSSB Line

FSSB (FANUC Serial Servo Bus) line is defined as follows.



The main board has two slots, "JGP1" and "JGP2" for auxiliary axis boards.
In one auxiliary axis board system, either JGP1 or JGP2 can be used for auxiliary axis board 1. By ordinary, please install the auxiliary axis board to JGP2.
In two auxiliary axis boards system, JGP2 is for auxiliary axis board 1 and JGP1 is for auxiliary axis board 2.

B.8.2 Setting 1 (FSSB line)

Please enter the number of FSSB line that added axis is connected to.

```
 ******** Group 2 Initialization ********
 ********** Independent Axes ***********

 -- FSSB configuration setting --
  1: FSSB line 1 (main axis card)
  2: FSSB line 2 (main axis card)
  3: FSSB line 3 (auxiliary axis board 1)
  5: FSSB line 5 (auxiliary axis board 2)
Select FSSB line >
Default value = 1
```

Please enter the value 1, 2, 3 or 5.

There are four FSSB lines, 1, 2, 3 and 5. Usually, the first FSSB line is used unless the system has many auxiliary axes or multi robots. This setting is necessary for extended axis and each motion groups, except for group 1.

B.8.3 Setting 2 (Number of total axes on FSSB line 1)

Please enter the number of total axes on FSSB line 1.
This setting is necessary only when added axis is on FSSB line 2.

```
Total number of axes on FSSB >
```

Please enter the number of total axes on FSSB line 1.
The number includes robot axes.

B.8.4 Setting 3 (Hardware start axis)

Please enter the start axis number of the first axis in the total hardware configuration of the system. This setting is necessary only when you set up the first axis in a motion group. It is not necessary when you add 2nd or later auxiliary axis in the group.

```
 ******** Group 2 Initialization ********
 ********** Independent Axes ***********

 -- Hardware start axis setting --
 Enter hardware start axis
 (Valid range: 1 - 32)
```

Please enter the number within valid range.

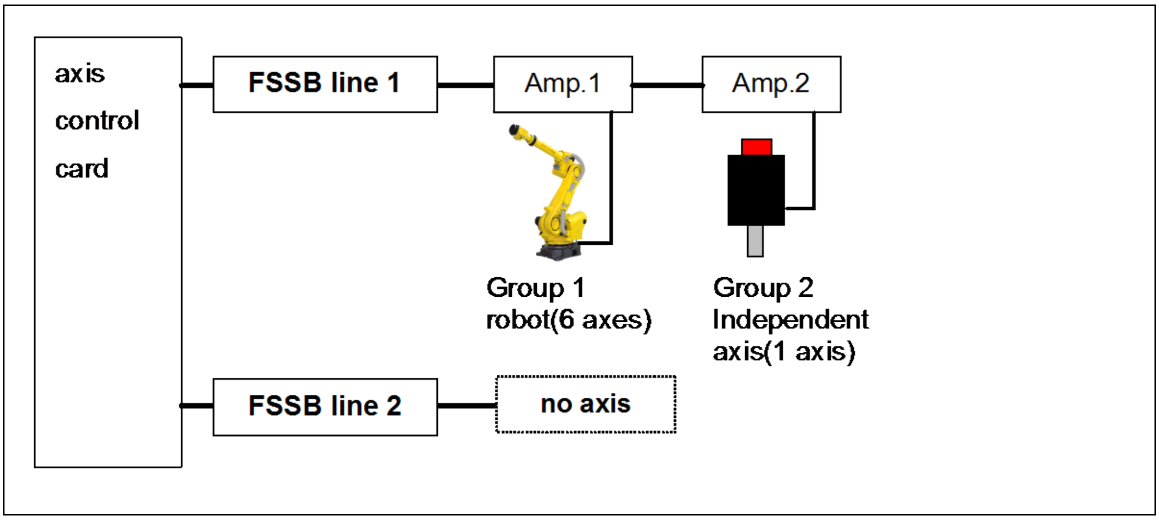
Valid range of hardware start axis differs according to the FSSB line to which auxiliary axis is connected.

FSSB line	Valid hardware start axis
1	7~32 (*1)
2	* ~36 (*2)
3	37~60 (*3)
5	61~84 (*4)

(*1) It is possible to use the number less than 7 when the robot does not have six axes.
(*2) The lower limit of hardware start axis for 2nd FSSB line differs according to the total axis number that is connected to the 1st FSSB line.
* = (Total axis number on the 1st FSSB line) + 1
(*3) The lower limit of hardware start axis for 3rd FSSB line is 37 irrespective of the number of axes on 1st and 2nd FSSB line.
(*4) The lower limit of hardware start axis for 4th FSSB line is 61 irrespective of the number of axes on 1st, 2nd and 3rd FSSB line.

B.8.5 Setup Examples

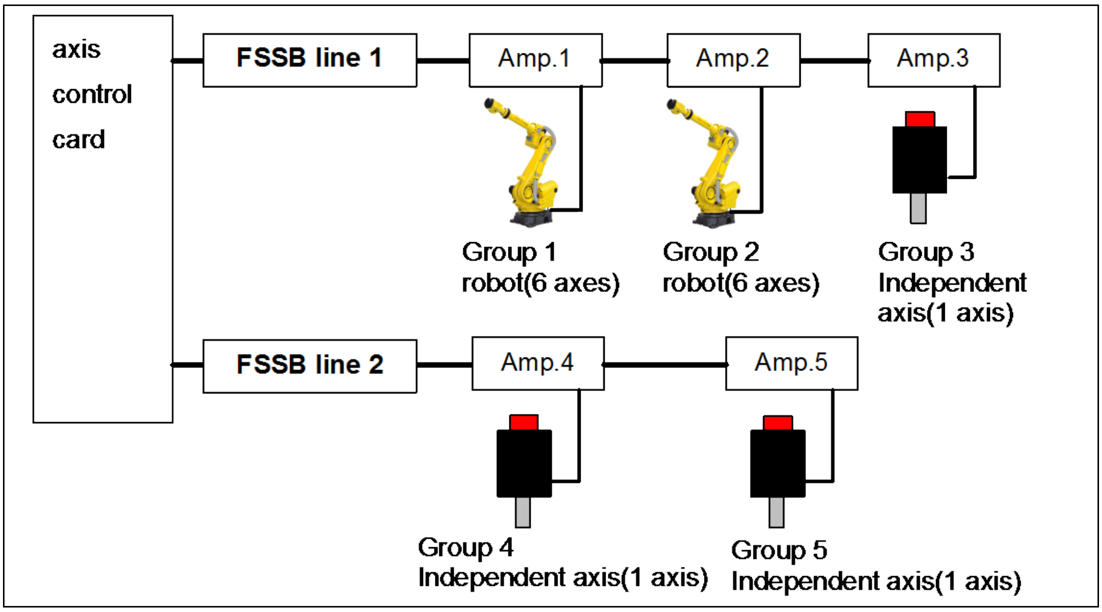
B.8.5.1 Example 1



When system configuration is like above figure, please set up as follows.

Motion group	FSSB line	Total number of axes on FSSB line 1	Hardware start axis
1	1	No need to set up	1
2	1	No need to set up	7

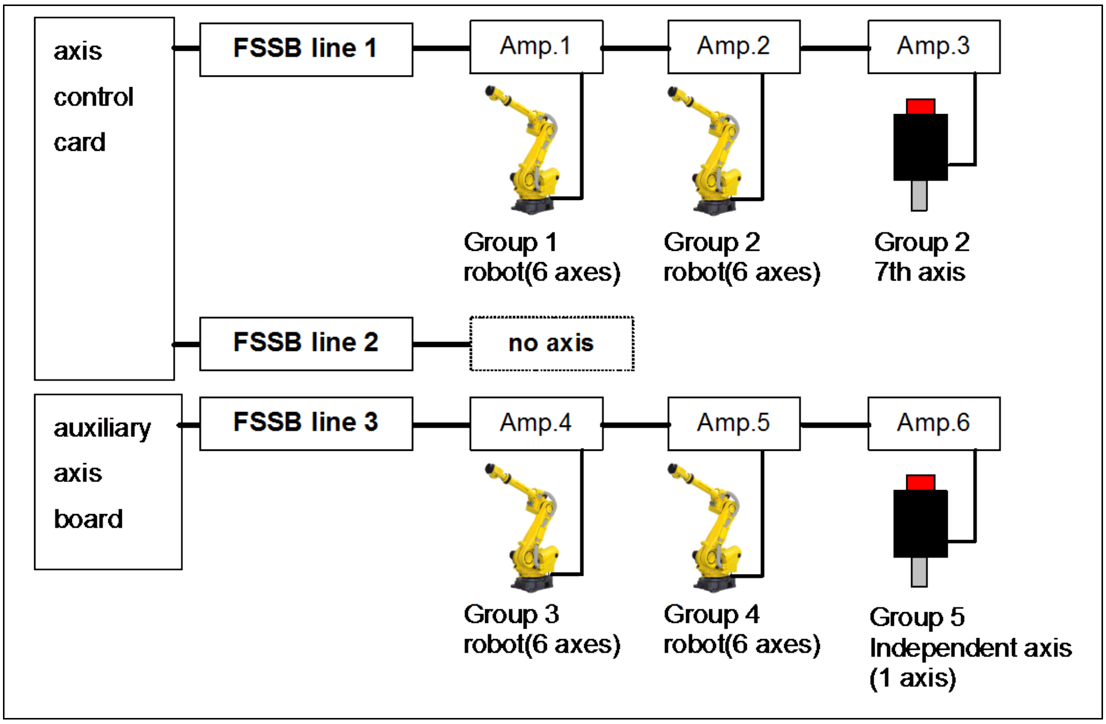
B.8.5.2 Example 2



When system configuration is like above figure, please set up as follows.

Motion group	FSSB line	Total number of axes on FSSB line 1	Hardware start axis
1	1	No need to set up	1
2	1	No need to set up	7
3	1	No need to set up	13
4	2	13	14
5	2	No need to change(13)	15

B.8.5.3 Example 3



When system configuration is like above figure, please set up as follows.

Motion group	FSSB line	Total number of axes on FSSB line 1	Hardware start axis
1	1	No need to set up	1
2	1	No need to set up	7
3	3	No need to set up	37(*)
4	3	No need to set up	43
5	3	No need to set up	49

(*) The number of hardware axis starts from 37 for 3rd FSSB line irrespective of the number of axes on 1st and 2nd FSSB line.
Hardware start axis of group 3 is not 14 but 37 for the above hardware configuration.

B.9 POSITIONER SETUP

Step

1 Turn on the controller with [PREV] and the [NEXT] key pressed. Then select [3 Controlled start].
2 Press [MENU] key and select “9. MAINTENANCE”.
3 The following screen will be displayed.

```
ROBOT MAINTENANCE
                                          1/10
    Setup Robot System Variables

      Group  Robot Library/Option   Ext Aces
        1    R-2000iB/165F             0
        2    Basic Positioner          0

   [ TYPE ]  ORD NO   AUTO   MANUAL
```

Press arrow (↑↓) keys and move the cursor to “Basic Positioner”. Then press F4, MANUAL.

4 “FSSB configuration setting” screen will be displayed. Please setup FSSB line and hardware start axis according to the instruction in section B.8. After adding at least one axis, you will no longer see this screen.
5 Select Kinematics Type.
Select “1: Known Kinematics” if the measurements of offset values between POSITONER axes are accurately known. If not, select “2: Unknown Kinematics”. After adding at least one axis, you will no longer see this screen.

```
    ******** Group % Initialization ********
    *********** Basic Positioner ***********

    -- Kinematics Type Setting --
    1: Known Kinematics
    2: Unknown Kinematics
    Select Kinematics Type?
```

6 The following screen will be displayed.

"%" is the group number.
"#" is the total number of axes installed. Initial value of # is 0.

```
******** Group % Initialization ********
*********** Basic Positioner ***********

Total Axes Installed = #
  1: Display/Modify Axis 1~4
  2: Add Axis
  3: Delete Axis
  4: Exit
  Select Item?
```

To add a POSITIONER axis, select "2. Add Axis". Then setup procedure starts.
To delete a POSITIONER axis, select "3. Delete Axis".
If the "3 Delete Axis" is selected, the following massage is displayed. "?" is the deleted axis number.

```
******** Group % Initialization ********
*********** Basic Positioner ***********

Total Axes Installed = #
  1: Display/Modify Axis 1~4
  2: Add Axis
  3: Delete Axis
  4: Exit
  Select Item? 3
POSITIONER Axis ? Was Deleted
Press ENTER to Continue
```

NOTE
Please refer to the mechanical specification sheet for the following procedure.

7 Select the motor of the axis.
(This example explains the procedure by the figures to set up first axis in the group2.)

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- MOTOR SELECTION --

 1: Standard Method
 2: Enhanced Method
 3: Direct Entry Method
Select ==>
```

If you select "1: Standard Method", then the following screen is displayed.
Select Motor Size.

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- MOTOR SIZE (alpha iS) --

 60. aiS2    64. aiS22
 61. aiS4    65. aiS30
 62. aiS8    66. aiS40
 63. aiS12
  0. Next Page
Select ==>
```

8 Select Motor Type.

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- MOTOR TYPE --

 1. /2000    11. /4000
 2. /3000    12. /5000
             13. /6000
Select ==>
```

⚠ CAUTION

The new model of α*i*/β*i* series servo motor (A06B-2□□□-Bxxx) has an additional character "-B" following the original model name such as "α*i*S8/4000-B". However, when you select the motor at this menu, select the servo motor excluding the additional character "-B". For example, in the case of "α*i*S8/4000-B", select "α*i*S8/4000" at this menu.

9 Select Current Limit for Motor.

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- CURRENT LIMIT FOR MOTOR --

 2.    4A   10.   20A
 5.   40A   12.  160A
 7.   80A

Select ==>
```

10 Enter Amplifier Number.

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Amplifier Number Setting --

Enter Amplifier Number (1~84)?
```

11 Select Amplifier Type.

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Amplifier Type Setting --

 1.  A06B-6400 series 6 axes amplifier
 2.  A06B-6240 series Alpha i amp. or
     A06B-6160 series Beta i amp.

Select Amplifier Type?
```

12 Select Axis Type.

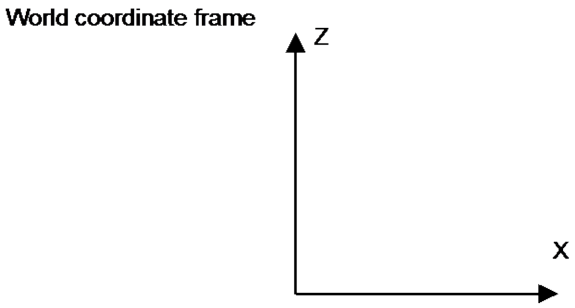
```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Axis Type Setting --

1: Linear Axis
2: Rotary Axis

Select Axis Type?
```

If the axis is linear axis, select “1: Linear Axis”.
If the axis is rotary axis, select “2: Rotary Axis”.

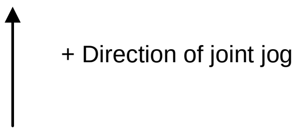
13 Select Direction of the axis. “Direction” is the axis direction in the world coordinate system. The direction has a sign (+/-).

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Direction Setting --

1: +X          3: +Y          5: +Z
2: -X          4: -Y          6: -Z

Select Direction?
```

Example)

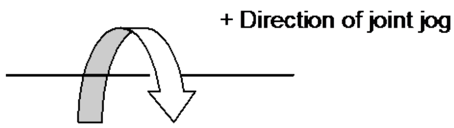


Linear axis



In this case, the direction should be set to “+Z”.

Rotary axis



In this case, the direction should be set to “+X”.   (Right-handed screw rule)

14 If you have select “Known Kinematics” in step 5, the following screens will be displayed. If the Kinematics is “Unknown Kinematics”, this procedure will be skipped.
Enter Offset value in X direction.

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Offset Setting --
Enter Offset X (mm)?
```

Enter Offset value in Y direction.

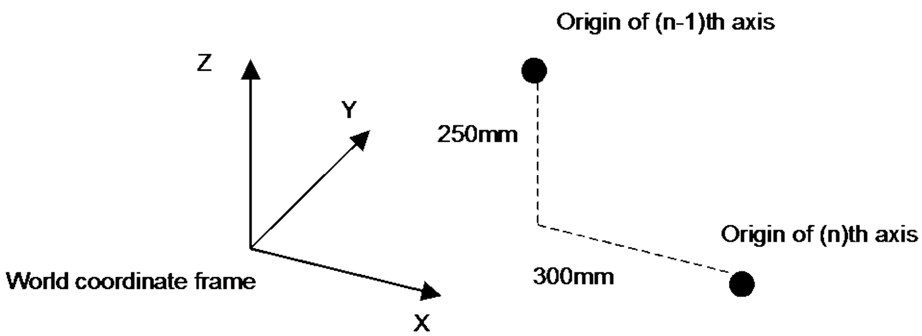
```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Offset Setting --
Enter Offset X (mm)?

Enter Offset Y (mm)?
```

Enter Offset value in Z direction.

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Offset Setting --
Enter Offset X (mm)?

Enter Offset Y (mm)?

Enter Offset Z (mm)?
```

For the 1st axis, enter offset values between the origin of the world coordinate and that of the axis. For the 2nd and later axes, enter offset values between the origin of the axis and that of the previous axis.

Example)



In this case, enter offset values for (n)the axis as follows.

Offset X: 300mm

Offset Y: 0mm
Offset Z: -250mm

15 Enter Gear Ratio.
For a linear axis, enter the distance of the motion which corresponds to one revolution of the motor. (UNIT: mm/mot-rev)

- Screen for a linear axis

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Get Ratio Setting --

Enter Gear Ratio (mm/mot-rev)?
```

For a rotary axis, enter the number of revolution of the motor which corresponds to one revolution of the axis. (UNIT: motor-rev/axis-rev)

- Screen for a rotary axis

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Get Ratio Setting --

Enter Gear Ratio (mot-rev/axs-rev)?
```

16 Enter Maximum Speed of the axis.

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Maximum Speed Setting --

Suggested Speed = 150.000 mm/sec
(Calculated with Max Motor Speed)

Enter (1: Change, 2: No Change)?
```

"Suggested Speed" is the default value.
If you want to change the value, select "1: Change", then, enter Max Speed.
If you want to use the suggested speed, select "2: No Change".

(In the case of changing Max Speed)

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Maximum Speed Setting --

Suggested Speed = 150.000 mm/sec
(Calculated with Max Motor Speed)

Enter (1: Change, 2: No Change)? 1
Enter Max Speed (mm/sec)?
```

17 Select Motion Sign.
Select “1: TRUE” if the joint coordinate position of this axis increases when the motor rotates in the plus direction.
Select “2: FALSE” if the joint coordinate position of this axis decreases when the motor rotates in the plus direction.

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Motion Sign Setting --

Current value = XXXXX
Enter (1: TRUE, 2: FALSE)?
```

18 Enter Upper Limit of the axis.
Unit for linear axis is “mm”.
Unit for rotary axis is “degree”.

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Upper Limit Setting --

Enter Upper Limit (deg)?
```

NOTE

You can determine Upper Limit of the POSITIONER axis by yourself. Here, there is a constraint on Lower Limit of this axis:

(Upper Limit) – (Lower Limit) = (Stroke Range)

Example)

Stroke Range = 100mm
Upper Limit = 50mm
Lower Limit = (Upper Limit) – (Stroke Range) = -50mm

19 Enter Lower Limit of the axis.
Unit for linear axis is “mm”.

Unit for rotary axis is “degree”.

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Lower Limit Setting --

Enter Lower Limit (deg)?
```

20 Enter the mastering position (Master Position) of the axis.

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Master Position Setting --

Enter Master Position (deg)?
```

21 Enter the acceleration/deceleration time constants (Accel Time).
First, enter the 1st accel time (Accel Time 1).

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Accel Time 1 Setting --

Current value = 256 msec
Enter (1: Change, 2: No Change)?
```

“Current value” is the default value.
If you want to change the value, select “1: Change”, then enter the new value.
Next, enter the 2nd accel time (Accel Time 2).

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Accel Time 2 Setting --

Current value = 128 msec
Enter (1: Change, 2: No Change)?
```

“Current value” is the default value.
If you want to change the value, select “1: Change”, then enter the new value.

22 Select use or not use the exponential acceleration/deceleration time filter.

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Exponential Filter Setting --

Exp Filter Valid = XXXXX
Enter (1: TRUE, 2: FALSE)?
```

If you use Exponential Filter, then enter the exponential acceleration/deceleration time constant.

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Exponential Accel Time Setting --

Current value =  20 msec
Enter (1: Change, 2: No Change)?
```

"Current value" is the default value.
If you want to change the value, select "1: Change", then enter the new value.

23 Enter Minimum Accel Time. When doing motion, if the original acceleration/deceleration time is smaller than the specified time, the acceleration/deceleration time will be clamped to the specified time.

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Minimum Accel Time Setting --

Current value = 384 msec
Enter (1: Change, 2: No Change)?
```

"Current value" is the default value.
If you want to change the value, select "1: Change", then enter the new value.

24 Enter Load Ratio. This value is the ratio of all load inertia to the rotor inertia. The valid range of Load Ratio is from 1.0 to 5.0 . If you don't set this value, enter "0".

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Load Ratio Setting --

            LoadInertia + MotorInertia
LoadRatio = --------------------------
                   MotorInertia

Enter Load Ratio (0:None, 1~5:Valid)?
```

25 Enter Brake Number. Enter Brake Number connected to the extend axis

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Brake Number Setting --

Enter Brake Number (0~32)?
```

26 Select the type of brake control (Servo Timeout).

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Servo Off Setting --

Servo Off Valid = XXXXX
Enter (1: TRUE, 2: FALSE)?
```

If you select “1: Enable”, then enter the delay time of brake control (Servo Off Time).

```
**** Group: 2 Axis: 1 Initialization ***
*********** Basic Positioner ***********

-- Servo Off Time Setting --

Enter Servo Off Time (0~30 sec)?
```

27 Come back to the screen of step 6.

```
    ******** Group 2 Initialization ********
    *********** Basic Positioner ***********

    Total Axes Installed = #
      1: Display/Modify Axis 1~4
      2: Add Axis
      3: Delete Axis
      4: Exit
      Select Item?
```

- To display/modify the POSITIONER axis setting, select "1: Display/Modify Axis".
- To add the next POSITIONER axis, select "2: Add Axis", then start the procedure from step 7.
- To delete the POSITIONER axis, select "3: Delete Axis".
- To finish POSITIONER axis setup, select "4: Exit".

> ⚠ **CAUTION**
> Positioner which has 5 or more axes cannot be used.

B.10 EXTENDED AXIS SETUP

Step

1 Turn on the controller with [PREV] and the [NEXT] key pressed. Then select [3 Controlled start].
2 Press [MENU] key and select "9. MAINTENANCE".
3 The following screen will be displayed.

```
ROBOT MAINTENANCE
                                             1/10
    Setup Robot System Variables

      Group  Robot Library/Option   Ext Aces
        1    R-2000iB/165F             0
             Extended Axis Control

   [ TYPE ]  ORD NO    AUTO    MANUAL
```

Press arrow (↑,↓) keys and move the cursor to "Extended Axis Control". Then press F4, MANUAL.

4 Select the group of the extended axis.

```
**** EXTENDED AXIS SETTING PROGRAM ****

-- SELECT GROUP --

 0. EXIT
 1. Group 1
```

5 “Hardware start axis setting” screen will be displayed. Please setup hardware start axis according to the instruction in section B.8. After adding at least one axis, you will no longer see this screen.

6 The following screen will be displayed.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 Initialization ******

                             E1 E2 E3
*** Group 1 Total Ext Axis = *  *  *
  1: Display/Modify Ext axis 1~3
  2: Add Ext axes
  3: Delete Ext axes
  4: EXIT
Select?
```

To add new extended axis, select “2. Add Ext axis”. Then setup procedure starts.

7 Enter the number of extended axis. For the first extended axis of the group, input “1”, for the second extended axis of the group, input ”2”, for the third axis of the group, input “3”. Please add axes in order of “1”, “2”, “3”.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 Initialization ******
                             E1 E2 E3
*** Group 1 Total Ext Axis = *  *  *
  1: Display/Modify Ext Axis 1~3
  2: Add Ext Axis
  3: Delete Ext Axis
  4: EXIT
Select? 2
Enter axis to add (1~3)?
```

8 Select the motor of the axis.
(This example explains the procedure by the figures to set up first axis in the group1.)

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- MOTOR SELECTION --

 1: Standard Method
 2: Enhanced Method
 3: Direct Entry Method
Select ==>
```

If you select “1: Standard Method”, then the following screen will be displayed.
Select Motor Size.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- MOTOR SIZE (alpha iS) --

 60. aiS2    64. aiS22
 61. aiS4    65. aiS30
 62. aiS8    66. aiS40
 63. aiS12
  0. Next Page
Select ==>
```

9 Select Motor Type.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- MOTOR TYPE --

 1. /2000    11. /4000
 2. /3000    12. /5000
             13. /6000
Select ==>
```

⚠ CAUTION

The new model of α*i*/β*i* series servo motor (A06B-2□□□-Bxxx) has an additional character "-B" following the original model name such as "α*i*S8/4000-B". However, when you select the motor at this menu, select the servo motor excluding the additional character "-B". For example, in the case of "α*i*S8/4000-B", select "α*i*S8/4000" at this menu.

10 Select Current Limit for Motor.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- CURRENT LIMIT FOR MOTOR --

 2.    4A   10.   20A
 5.   40A   12.  160A
 7.   80A

Select ==>
```

11 Select the type of extended axis from among the four types.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- EXTENDED AXIS TYPE --

  1. Integrated Rail (Linear axis)
  2. Integrated Arm  (Rotary axis)
  3. Auxiliary Linear Axis
  4. Auxiliary Rotary Axis

Select?
```

NOTE

1 Integrated Rail / Arm:
World coordinate frame is fixed to the ground. Cartesian position of robot TCP varies if this extended axis moves.
Auxiliary Linear / Rotary Axis:
World coordinate frame is fixed to the robot. Cartesian position of robot TCP does not vary if this extended axis moves.

2 Do the following setting according to the selected extended axis type.

Integrated Rail (Linear axis):

Enter the axis direction in the world coordinate system.
The displayed screens differ between software version 7DC3/05 or older and 7DC3/06 or later. On software version 7DC3/06 or later, diagonal axis on XY plane can be set.

(In the case of software version 7DC3/05 or older)

In the screen below, select the direction of the extended axis in the world coordinate frame.
The positive direction of the axis movement matches the positive direction of the X, Y, or Z axis of the world coordinate frame. It cannot be changed to the negative direction.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

  1. Integrated Rail (Linear axis)
  2. Integrated Arm  (Rotary axis)
  3. Auxiliary Linear Axis
  4. Auxiliary Rotary Axis

Select? 1
Direction 1:X  2:Y  3:Z
Enter Direction (1~3)?
Select?
```

(In the case of software version 7DC3/06 or later)

In the screen below, select the direction the extended axis in the world coordinate frame.
If the direction of the extended axis matches the negative direction of the X or Y of the world coordinate frame, select the diagonal axis described below.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *
-- DIRECTION SETTING –
1. + X axis
2. + Y axis
3. + Z axis
4. Diagonal axis on XY plane
Select Direction (1 – 4)->
```

When the extended axis is a diagonal axis (Linear axis is along the diagonal direction on XY plane), select “4. Diagonal axis on XY plane”. When you select it, the following screen is displayed and you can enter an azimuth angle. Azimuth angle is an offset angle with respect to +X axis in the world coordinate frame. You can set between -180 and 180[deg].

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *
-- AZIMUTH ANGLE SETTING --
 90 [deg]: + Y axis
  0 [deg]: + X axis
-90 [deg]: - Y axis
Set azimuth angle (-180 – 180[deg])->
```

⚠ CAUTION

The following function cannot be used with diagonal axis.

- Online position modification, Program Shift Function, Mirror Shift Function, Angle Entry Shift Function (Standard Function)
- Rail Tracking (Part of Line Tracking.)
- Deadlock Prevention Function (Part of Intelligent Interference Check.)
- RailZone Function (Part of Intelligent Interference Check.)

Integrated Arm (Rotary axis):

Enter the offset length in Z direction from the extended axis frame origin to the robot world frame origin.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- EXTENDED AXIS TYPE --

  1. Integrated Rail (Linear axis)
  2. Integrated Arm  (Rotary axis)
  3. Auxiliary Linear Axis
  4. Auxiliary Rotary Axis

Select? 2
Enter Offset length (mm)?
```

Next, enter the arm length of extended axis.

- [Case 1] Direction of the extended axis is parallel to X-axis of the robot world frame:
  Enter the offset length in Y direction from the extended axis frame origin to the robot world frame origin.
- [Case 2] Direction of the extended axis is parallel to Y-axis of the robot world frame:
  Enter the offset length in X direction from the extended axis frame origin to the robot world frame origin.
- [Case 3] Direction of the extended axis is parallel to Z-axis of the robot world frame:
  Enter the offset length in X direction from the extended axis frame origin to the robot world frame origin.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *
-- EXTENDED AXIS TYPE --

  1. Integrated Rail (Linear axis)
  2. Integrated Arm  (Rotary axis)
  3. Auxiliary Linear Axis
  4. Auxiliary Rotary Axis

Select? 2
Enter Offset length (mm)?
Enter Arm Length (mm)?
```

NOTE

In Case 3, direction of the extended axis and X-axis of the robot world frame must intersect. In other words, the direction of extended axis must be on Y = 0 plane of the robot world frame.

Then, select the direction of the extended axis. The extended axis should be parallel to one of the robot world frame (X,Y,Z). Select one.
The positive rotation direction of the entered world coordinate frame axis matches the positive rotation direction of extended axis (See the figure in the step 13 of chapter B.9 for positive rotation direction). It cannot be changed to the negative direction.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *
  2. Integrated Arm  (Rotary axis)
  3. Auxiliary Linear Axis
  4. Auxiliary Rotary Axis

Select? 2
Enter Offset length (mm)?
Enter Arm Length (mm)?
Direction 1:X  2:Y  3:Z
Enter Direction (1~3)?
Select?
```

12 Enter Gear Ratio.
For a linear axis, enter the distance of the motion which corresponds to one revolution of the motor. (UNIT: mm/mot-rev)

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- GEAR RATIO --

  For a linear axis it is the number of
  mm's traveled for one rotation of the
  motor.
Enter Gear Ratio?
```

For a rotary axis, enter the number of revolution of the motor which corresponds to one revolution of the axis. (UNIT: motor-rev/axis-rev)

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- GEAR RATIO --

Enter Gear Ratio
```

13 Enter Max Joint Speed of the axis.
"Suggested Speed" is the default value.
If you want to change the value, select "1: Change".Then, enter Max Joint Speed.
If you want to use "Suggested Speed", select "2: No Change".

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- MAX JOINT SPEED SETTING --

Suggested Speed =  150.000(mm/sec)
(Calculated with Max motor speed)

Enter (1:Change, 2:No Change)?
```

(In the case of changing Max Joint Speed)

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- Max JOINT SPEED SETTING --

Suggested Speed =  150.000(mm/sec)
(Calculated with Max motor speed)

Enter (1:Change, 2:No Change)? 1
Enter Max Speed  (mm/s)
```

14 Select Motor Direction.
Select “1: TRUE” if the joint coordinate position of this axis increases when the motor rotates in the plus direction.
Select “2: FALSE” if the joint coordinate position of this axis decreases when the motor rotates in the plus direction.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- MOTOR DIRECTION --

Ext_axs 1 Motion Sign = XXXXX
Enter (1: TRUE, 2:FALSE)?
```

15 Enter Upper Limit of the axis.
Unit for linear axis is “mm”.
Unit for rotary axis is “degree”.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- UPPER LIMIT --

Enter Upper Limit (deg)?
```

NOTE

You can determine Upper Limit of the extended axis by yourself. Here, there is a constraint on Lower Limit of this axis:

(Upper Limit) – (Lower Limit) = (Stroke Range)

Example)

Stroke Range = 100mm
Upper Limit = 50mm
Lower Limit = (Upper Limit) – (Stroke Range) = -50mm

16 Enter Lower Limit of the axis.
Unit for linear axis is “mm”.
Unit for rotary axis is “degree”.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- LOWER LIMIT --

Enter Lower Limit (deg)?
```

17 Enter the mastering position (Master Position) of the axis.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- MASTER POSITION --

Enter Master Position (deg)?
```

18 Enter the acceleration/deceleration time constants (ACC/DEC Time).
First, enter the 1st ACC/DEC Time (acc_time1).
“Default acc_time1” is the default value.
If you want to change the value, select “1: Change”, then enter the new value.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- ACC/DEC TIME --

Default acc_time1 = 256(ms)
Enter (1:Change, 2:No Change)?
```

Next, enter the 2nd ACC/DEC Time (acc_time2).
“Default acc_time2” is the default value.
If you want to change the value, select “1: Change”, then enter the new value.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- ACC/DEC TIME --

Default acc_time1 = 256(ms)
Enter (1:Change, 2:No Change)?

Default acc_time2 = 128(ms)
Enter (1:Change, 2:No Change)?
```

19 Enter Minimum Accel Time. When doing motion, if the calculated acceleration/deceleration time is smaller than the specified time, the acceleration/deceleration time will be clamped to the specified time. “Default min_acctime” is the default value.
If you want to change the value, select “1: Change”, then enter the new value.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- MIN_ACCEL TIME --

Default min_acctime = 128(ms)
Enter (1:Change, 2:No Change)?
```

20 Enter Load Ratio. This value is the ratio of all load inertia to the rotor inertia. The valid range of Load Ratio is from 1.0 to 5.0 . If you don’t set this value, enter “0”.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- LOAD RATIO --

            LoadInertia + MotorInertia
LoadRatio = --------------------------
                   MotorInertia

Enter Load ratio? (0:None, 1~5:Valid)
```

21 Enter Amplifier Number.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- SELECT AMP NUMBER --

Enter amplifier Number (1~84)?
```

22 Select Amplifier Type.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- SELECT AMP TYPE --

 1.  A06B-6400 series 6 axes amplifier
 2.  A06B-6240 series Alpha i amp. or
     A06B-6160 series Beta i amp.

Select?
```

23 Enter Brake Number. Enter Brake Number connected to the extend axis.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- BREAKE SETTING --

Enter Brake Number (0~32)?
```

24 Select the type of brake control. If you want to use brake control, select “1: Enable”, then enter the delay time of brake control (Servo Off Time).If you don’t use, select “2: Disable”.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- SERVO TIMEOUT --

Servo Off is XXXXX
Enter (1: Enable  2: Disable)?
```

In case of selecting “1: Enable”.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization *

-- SERVO TIMEOUT VALUE --

Enter Servo Off Time? (0.0~30.0 Sec)
```

25 Come back to the screen of step 6.

```
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 Initialization ******

                               E1 E2 E3
*** Group 1 Total Ext Axis = I  *  *
  1: Display/Modify Ext axis 1~3
  2: Add Ext axes
  3: Delete Ext axes
  4: EXIT
Select?
```

- To display/modify the extended axis setting, select "1: Display/Modify Ext axis".
- To add the next extended axis, select "2: Add Ext Axes", then start the procedure from step 7.
- To delete the extended axis, select "3: Delete Ext axes".
- To finish extended axis setup, select "4: EXIT".

B.11 INDEPENDENT ADDITIONAL AXIS SETUP

Step

1 Turn on the controller with [PREV] and the [NEXT] key pressed. Then select [3 Controlled start].
2 Press [MENU] key and select "9. MAINTENANCE".
3 The following screen will be displayed.

```
ROBOT MAINTENANCE
                                          1/10
    Setup Robot System Variables

      Group  Robot Library/Option   Ext Aces
        1    R-2000iB/165F             0
        2    Independent Axes          0

 [ TYPE ]  ORD NO   AUTO   MANUAL
```

Press arrow (↑↓) keys and move the cursor to "Independent Axes". Then press F4, MANUAL.

4 "FSSB configuration setting" screen will be displayed. Please setup FSSB line and hardware start axis according to the instruction in section B.8. After adding at least one axis, you will no longer see this screen.
5 The following screen will be displayed.
"%" is the group number.
"#" is the total number of the independent axes installed. Initial value of # is 0.

```
******** Group % Initialization ********
********** Independent Axes ************

Total Axes Installed = #
  1: Display/Modify Axis 1~4
  2: Add Axis
  3: Delete Axis
  4: Exit
  Select Item?
```

To add an independent axis, select “2. Add Axis”. Then setup procedure starts.
To delete an independent axis, select “3. Delete Axis”.

Please refer to the mechanical specification sheet for the following procedure.

6 Select the motor of the axis.
(This example explains the procedure by the figures to set up first axis in the group2.)

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- MOTOR SELECTION --

 1: Standard Method
 2: Enhanced Method
 3: Direct Entry Method
Select ==>
```

If you select “1: Standard Method”, then the following screen is displayed.
Select Motor Size.

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- MOTOR SIZE (alpha iS) --

 60. aiS2    64. aiS22
 61. aiS4    65. aiS30
 62. aiS8    66. aiS40
 63. aiS12
  0. Next Page
Select ==>
```

7 Select Motor Type.

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- MOTOR TYPE --

 1. /2000    11. /4000
 2. /3000    12. /5000
             13. /6000
Select ==>
```

⚠ CAUTION

The new model of α*i*/β*i* series servo motor (A06B-2□□□-Bxxx) has an additional character "-B" following the original model name such as "α*i*S8/4000-B". However, when you select the motor at this menu, select the servo motor excluding the additional character "-B". For example, in the case of "α*i*S8/4000-B", select "α*i*S8/4000" at this menu.

8 Select Current Limit for Motor.

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- CURRENT LIMIT FOR MOTOR --

 2.    4A   10.   20A
 5.   40A   12.  160A
 7.   80A

Select ==>
```

9 Select Axis Type.

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- INDEPENDENT AXES TYPE --

  1. Linear Axis
  2. Rotary Axis

Select?
```

If the axis does linear motion, select “1: Linear Axis”.
If the axis does rotary motion, select “2: Rotary Axis”.

10 Enter Gear Ratio.
For a linear axis, enter the distance of the motion which corresponds to one revolution of the motor. (UNIT: mm/mot-rev)

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- GEAR RATIO --

  For a linear axis it is the number of
  mm's traveled for one rotation of the
  motor.
Enter Gear Ratio?
```

For a rotary axis, enter the number of revolution of the motor which corresponds to one revolution of the axis. (UNIT: motor-rev/axis-rev)

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- GEAR RATIO --

Enter Gear Ratio
```

11 Enter Max Joint Speed of the axis.

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- MAX JOINT SPEED SETTING --

Suggested Speed = 150.000 mm/sec
(Calculated with Max Motor Speed)

Enter (1:Change, 2:No Change)?
```

“Suggested Speed” is the default value.
If you want to change the value, select “1: Change”.
Then, enter Max Joint Speed.

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- MAX JOINT SPEED SETTING --

Suggested Speed = 150.000 mm/sec
(Calculated with Max Motor Speed)

Enter (1:Change, 2:No Change)? 1
Enter Max Speed  (mm/s)?
```

12 Select Motor Direction.
Select “1: TRUE” if the joint coordinate position of this axis increases when the motor rotates in the plus direction.
Select “2: FALSE” if the joint coordinate position of this axis decreases when the motor rotates in the plus direction.

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- MOTOR DIRECTION --

INDEPENDENT AXES 1 Motion Sign = XXXXX
Enter (1: TRUE, 2:FALSE)?
```

13 Enter Upper Limit of the axis.
Unit for linear axis is “mm”.
Unit for rotary axis is “degree”.

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- UPPER LIMIT --

Enter Upper Limit (deg)?
```

NOTE

You can determine Upper Limit of the independent axis by yourself. Here, there is a constraint on Lower Limit of this axis:

(Upper Limit) – (Lower Limit) = (Stroke Range)

Example)

Stroke Range = 100mm
Upper Limit = 50mm
Lower Limit = (Upper Limit) – (Stroke Range) = -50mm

14 Enter Lower Limit of the axis.
Unit for linear axis is “mm”.
Unit for rotary axis is “degree”.

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- LOWER LIMIT --

Enter Lower Limit (deg)?
```

15 Enter the mastering position (Master Position) of the axis.

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- MASTER POSITION --

Enter Master Position (deg)?
```

16 Enter the acceleration/deceleration time constants (ACC/DEC Time).
First, enter the 1st ACC/DEC Time (acc_time1).
"Default acc_time1" is the default value.
If you want to change the value, select "1: Change", then enter the new value.

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- ACC/DEC TIME --

Default Value of acc_time1 = 256 msec
Enter (1:Change, 2:No Change)?
```

Next, enter the 2nd ACC/DEC Time (acc_time2).
"Default acc_time2" is the default value.
If you want to change the value, select "1: Change", then enter the new value

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- ACC/DEC TIME --

Default Value of acc_time1 = 256 msec
Enter (1:Change, 2:No Change)?

Default Value of acc_time2 = 128 msec
Enter (1:Change, 2:No Change)?
```

17 Enter Minimum Accel Time. When doing motion, if the calculated acceleration/deceleration time is smaller than the specified time, the acceleration/deceleration time will be clamped to the specified time.
"Default min_acctime" is the default value.
If you want to change the value, select "1: Change", then enter the new value.

```
**** Group: 2 Axis:  1 Initialization ***
*********** Independent Axes ************

-- MIN_ACCEL TIME --

Default Value of min_acctime = 384 msec
Enter (1:Change, 2:No Change)?
```

18 Enter Load Ratio. This value is the ratio of all load inertia to the rotor inertia. The valid range of Load Ratio is from 1.0 to 5.0. If you don't set this value, enter "0".

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- LOAD RATIO --

            LoadInertia + MotorInertia
LoadRatio = --------------------------
                   MotorInertia

Enter Load ratio? (0:None 1~5:Valid)
```

19 Enter Amplifier Number.

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- SELECT AMP NUMBER --

Enter amplifier number (1~84)?
```

20 Select Amplifier Type.

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- SELECT AMP TYPE --

 1.  A06B-6400 series 6 axes amplifier
 2.  A06B-6240 series Alpha i amp. or
     A06B-6160 series Beta i amp.

Select?
```

21 Enter Brake Number. Enter Brake Number connected to the extend axis

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- BRAKE SETTING --

Enter Brake Number (0~32)?
```

22 Select the type of brake control (Servo Timeout). If you want to use brake control, select “1: Enable”, then enter the delay time of brake control (Servo Off Time).If you don’t use, select “2: Disable”.

```
**** Group: ? Axis: 1 Initialization ***
********** Independent Axes ************

-- SERVO TIMEOUT --

Servo Off is XXXXX
Enter (1: Enable  2: Disable)?
```

In case of selecting “1: Enable”.

```
**** Group: 2 Axis: 1 Initialization ***
********** Independent Axes ************

-- SERVO TIMEOUT VALUE --

Enter Servo Off Time? (0.0~30.0)
```

23 Come back to the screen of step 5.

```
******** Group 2 Initialization ********
********** Independent Axes ************

Total Axes Installed = #
  1: Display/Modify Axis 1~4
  2: Add Axis
  3: Delete Axis
  4: Exit
  Select Item?
```

- To display/modify the independent axis setting, select “1: Display/Modify Axis”.
- To add the next independent axis, select “2: Add Axis”, then start the procedure from step 6.
- To delete the independent axis, select “3: Delete Axis”.
- To finish independent axis setup, select “4: Exit”.

> ⚠ **CAUTION**
> Independent axis which has 5 or more axes cannot be used.

Operation of setting extend axis is now complete.

B.12 TP FIRMWARE

When the teach pendant is replaced, please update the firmware of the teach pendant (TP firmware). The TP firmware version is checked by performing a controlled start and then performing a cold start. A firmware update is performed when the TP firmware is older than the software version or different from the software series of the controller.

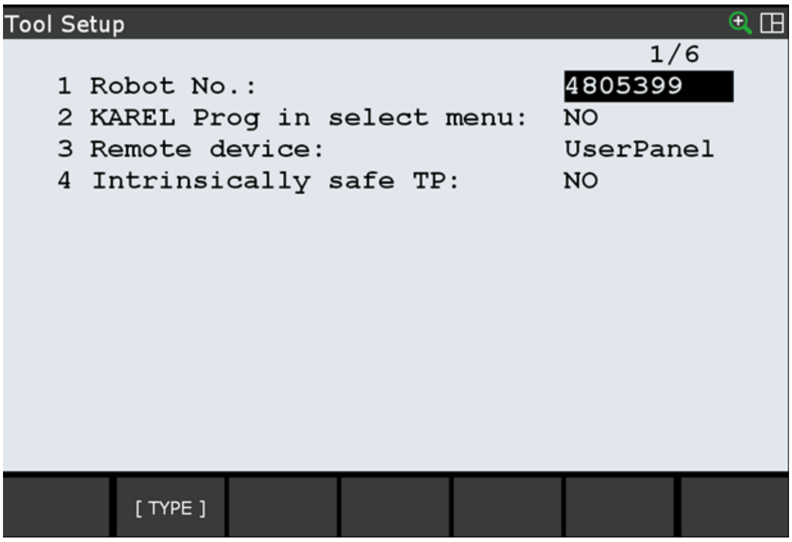
NOTE
1 If the software version is 7DC1/06(V8.10P/06) or earlier or the FROM is less than 64MB, the above method is not used. In this case, please contact your local FANUC representative.
2 Refer to Appendix B.1 for the details of controlled start and cold start.

The operator can check the TP firmware version by the following procedure.
1 Press the [MENU] key to display the screen menu.
2 Press “0 -- NEXT --”.
3 Select “4 STATUS” -> “2 Version ID”.
4 “TP Core Firmware” indicates the TP firmware version.

Procedure B-16 TP firmware update

Step

1 With the [PREV] key and the [NEXT] key held down on the teach pendant, turn on the controller. The configuration menu will be displayed.
2 Select “3. Controlled start”. The following setting screen for the controlled start menu will be displayed.



3 Press the [FCTN] key. A menu will be displayed. From that menu, select “1 START (COLD)”.
4 The TP firmware update starts.

⚠ CAUTION
The update takes a few minutes. Please do not turn off the controller while the update is performed.

5 A cold start is performed after the update is completed.

C SYSTEM VARIABLES

This part of this manual describes the names, functions, standard settings, and valid ranges of system variables.

Contents of this appendix

C.1 FORMAT OF A SYSTEM VARIABLE TABLE

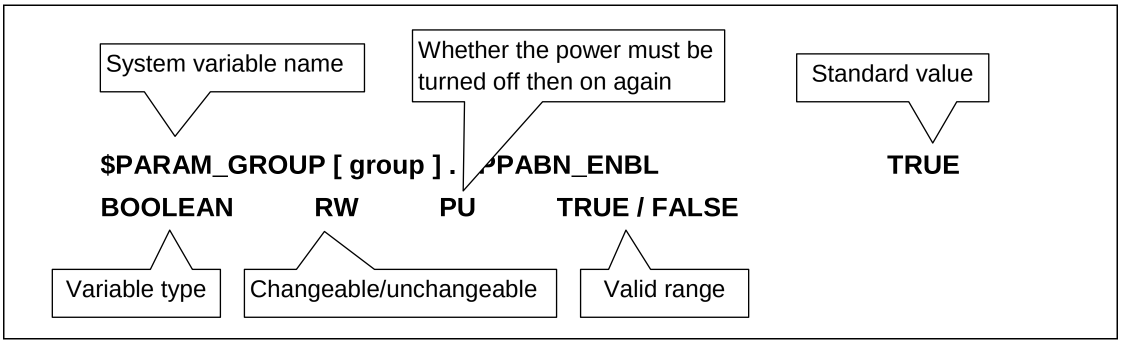


Table C.1 Format of a system variable table

System variable name		
Standard value	*	Intrinsic value for each model
Variable type	BOOLEAN	True/false type (TRUE/FALSE)
	BYTE	Integer (0 to 255)
	SHORT	Integer (-32768 to 32767)
	INTEGER	Integer (-1000000 to 1000000)
	REAL	Real number (-10000000000 to 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 C-1 Setting a system variable

Step

1. Press [MENU] key.
2. Select 0 (NEXT), then select 6 (SYSTEM).
3. Press the F1, [TYPE] key.
4. Select Variables. Then, the system variable screen will be displayed.

```
SYSTEM Variables
                                         1/638
      1 $AAVM              AAVM_T
      2 $ABSPOS_GRP        ABSPOS_GRP_T
      3 $ACC_MAXLMT        150
      4 $ACC_MINLMT        0
      5 $ACC_PRE_EXE       0
      6 $ALM_IF            ALM_IF_T
      7 $ANGTOL            [9] of REAL
      8 $APPLICATION       [9] of STRING[21]
      9 $AP_ACTIVE         6
     10 $AP_AUTOMODE       FALSE
     11 $AP_CHGAPONL       TURE

   [ TYPE ]  DETAIL
```

5 To change the value of a system variable, move the cursor to a desired item, enter a new value, then press the [ENTER] key or select a desired item by pressing the corresponding function key.

6 When a system variable contains multiple system variables, move the cursor to a desired item and press the [ENTER] key. Then, the low-order system variables are displayed.

```
SYSTEM Variables
                                       329/638
    327 $PADJ_SCHNUM       10
    328 $PARAM2_GRP        MRR2_GRP_T
    329 $PARAM_GROUP       MRR_GRP_T
    330 $PARAM_MENU        [21] of STRING[21]
    331 $PASSNAME          [10] of PASSNAME_T
    332 $PASSSUPER         PASSNAME_T
    333 $PASSWORD          PASSWORD_T
    334 $PAUSE_PROG        *uninit*
    335 $PCCRT             0
    336 $PCCRT_HOST        'PCCRT'
    337 $PCTP              0

   [ TYPE ]  DETAIL
```

```
SYSTEM Variables
    $PARAM_GROUP[1]                      1/236
       1 $BELT_ENABLE      FALSE
       2 $CART_ACCEL1      800
       3 $CART_ACCEL2      400
       4 $CIRC_RATE        1
       5 $CONTAXISNUM      0
       6 $EXP_ENBL         FALSE
       7 $JOINT_RATE       1
       8 $LINEAR_RATE      1
       9 $PATH_ACCEL1      800
      10 $PATH_ACCEL2      400
      11 $PROCESS_SPD      2000.000

   [ TYPE ]                    TRUE     FALSE
```

7 After changing the setting of the system variable for which PU is specified, cycle power of the controller. (PU is specified for all $PARAM_GROUP system variables.)

> **NOTE**
> The setting of a system variable for which RO (unchangeable) is specified cannot be changed.

C.2 SYSTEM VARIABLES

Power failure recovery

$SEMIPOWERFL FALSE
BOOLEAN RW TRUE/FALSE

[Function] Enables or disables the hot start

[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.

Break control

$PARAM_GROUP[group] . $SV_OFF_ENB[1] *
$PARAM_GROUP[group] . $SV_OFF_ENB[2] *
$PARAM_GROUP[group] . $SV_OFF_ENB[3] *
$PARAM_GROUP[group] . $SV_OFF_ENB[4] *
$PARAM_GROUP[group] . $SV_OFF_ENB[5] *
$PARAM_GROUP[group] . $SV_OFF_ENB[6] *
$PARAM_GROUP[group] . $SV_OFF_ENB[7] *
$PARAM_GROUP[group] . $SV_OFF_ENB[8] *
$PARAM_GROUP[group] . $SV_OFF_ENB[9] *
BOOLEAN RW PU TRUE / FALSE

[Function] Enables or disables the break control function

[Description] Specifies how the brake is applied when the axis doesn’t move for a given length of time. However the brake control function is disabled for all axes on the same break release DO if there is at least one axis that the brake control is disabled on the DO.

TRUE: It puts on brake when the axis doesn’t move for a given length of time and it puts off the brake when the axis starts to move.

FALSE: It never put on brake even if the axis doesn’t move for a given length of time.

$PARAM_GROUP[group] . $SV_OFF_TIME[1] *
$PARAM_GROUP[group] . $SV_OFF_TIME[2] *
$PARAM_GROUP[group] . $SV_OFF_TIME[3] *
$PARAM_GROUP[group] . $SV_OFF_TIME[4] *
$PARAM_GROUP[group] . $SV_OFF_TIME[5] *
$PARAM_GROUP[group] . $SV_OFF_TIME[6] *
$PARAM_GROUP[group] . $SV_OFF_TIME[7] *
$PARAM_GROUP[group] . $SV_OFF_TIME[8] *
$PARAM_GROUP[group] . $SV_OFF_TIME[9] *
INTEGER RW PU 0 to 100000000 (msec)

[Function] Time that elapses before the brake control function performs

[Description] Specifies the elapsed time [msec] before it puts on the brake by the brake control function.

$PARAM_GROUP[group] . $SV_OFF_ALL TRUE
BOOLEAN RW PU TRUE / FALSE

[Function] Enables or disables the synchronous brake control for all axes

[Description] Specifies how the brakes for all axes are applied at the same time.

TRUE: It puts on/off brakes for all axes at same time, i.e. it does not put on all brakes till all axes finish to move and it puts off all brakes when one axis starts to move.

FALSE: It puts on/off brakes independently, i.e. it puts on each brake release DO which all axes on the DO finish to move and it never put off each brake release DO which any axis on the DO doesn’t start to move. However brakes on same brake release

DO are not able to be put on/off independently because brakes for all axes on same brake release DO are controlled in a lump.

Mastering

$MASTER_ENB 0
ULONG RW 1 / 0

[Function] Displays positioning screen

[Description] When this variable is enabled, the positioning screen [6 SYSTEM, Master/Cal] is displayed on the teach pendant.

0: Positioning screen not displayed.
1: Positioning screen displayed.

$DMR_GRP[group]. $MASTER_DONE TRUE
BOOLEAN RW TRUE / FALSE

[Function] Indicates if mastering is completed.

[Description] Indicates if mastering has been completed.

[Setting] On the positioning screen [6 SYSTEM, Master/Cal]

$DMR_GRP[group]. $MASTER_COUN[1] *
$DMR_GRP[group]. $MASTER_COUN[2] *
$DMR_GRP[group]. $MASTER_COUN[3] *
$DMR_GRP[group]. $MASTER_COUN[4] *
$DMR_GRP[group]. $MASTER_COUN[5] *
$DMR_GRP[group]. $MASTER_COUN[6] *
$DMR_GRP[group]. $MASTER_COUN[7] *
$DMR_GRP[group]. $MASTER_COUN[8] *
$DMR_GRP[group]. $MASTER_COUN[9] *
INTEGER RW 0 to 100000000 (pulse)

[Function] Store mastering pulse counts

[Description] Pulsecoder count at zero degree position is stored. This value is calculated from current count at mastering and current position.

$DMR_GRP[group]. $GRAV_MAST *
INTEGER RW -1 to 1

[Function] Indicates if Mastering counts are taken with Gravity Compensation or not.

[Description] Indicates whether mastering counts are derived from Mastering with Gravity Compensation enabled or not. For details, refer to the chapter “Gravity Compensation” in “FANUC Robot series Optional Function OPERATOR’S MAUNAL” (B-83284EN-2).

1 Current mastering counts are derived from Mastering with Gravity Compensation.
0 Current mastering counts are derived from Mastering without Gravity Compensation.
-1 Unknown (not set).

$PARAM_GROUP[group]. $MASTER_POS[1] *
$PARAM_GROUP[group]. $MASTER_POS[2] *
$PARAM_GROUP[group]. $MASTER_POS[3] *
$PARAM_GROUP[group]. $MASTER_POS[4] *
$PARAM_GROUP[group]. $MASTER_POS[5] *
$PARAM_GROUP[group]. $MASTER_POS[6] *
$PARAM_GROUP[group]. $MASTER_POS[7] *
$PARAM_GROUP[group]. $MASTER_POS[8] *
$PARAM_GROUP[group]. $MASTER_POS[9] *
REAL RW PU -100000 to 100000 (deg)

[Function] Store jig position for jig mastering

[Description] Jig position for jig mastering is stored. Mastering pulse count is calculated from this data.

Quick mastering

$DMR_GRP[group]. $REF_DONE FALSE
BOOLEAN RW TRUE / FALSE

[Function] Indicates if setting of the reference point for quick mastering is completed.
[Description] When the reference point of quick mastering is set, the Pulsecoder count and coordinate values of the reference position are stored.
[Setting] On the positioning screen [6 SYSTEM, Master/Cal]

$DMR_GRP[group]. $REF_COUNT[1] 0
$DMR_GRP[group]. $REF_COUNT[2] 0
$DMR_GRP[group]. $REF_COUNT[3] 0
$DMR_GRP[group]. $REF_COUNT[4] 0
$DMR_GRP[group]. $REF_COUNT[5] 0
$DMR_GRP[group]. $REF_COUNT[6] 0
$DMR_GRP[group]. $REF_COUNT[7] 0
$DMR_GRP[group]. $REF_COUNT[8] 0
$DMR_GRP[group]. $REF_COUNT[9] 0
INTEGER RW 0 to 100000000 (pulse)

[Function] Store reference point mastering count
[Description] Store the count of the Pulsecoder when the robot is positioned at the reference point.

$DMR_GRP[group]. $REF_POS[1] 0
$DMR_GRP[group]. $REF_POS[2] 0
$DMR_GRP[group]. $REF_POS[3] 0
$DMR_GRP[group]. $REF_POS[4] 0
$DMR_GRP[group]. $REF_POS[5] 0
$DMR_GRP[group]. $REF_POS[6] 0
$DMR_GRP[group]. $REF_POS[7] 0
$DMR_GRP[group]. $REF_POS[8] 0
$DMR_GRP[group]. $REF_POS[9] 0
REAL RW -100000 to 100000 (rad)

[Function] Store reference point to be set during quick mastering
[Description] Store the reference point to be set during quick mastering.

Calibration

$MOR_GRP[group]. $CAL_DONE TRUE
BOOLEAN RW TRUE / FALSE

[Function] Indicates if calibration is completed.
[Description] To check the current position of the robot, the count of the Pulsecoder issued and the current position is calculated using mastering count. This check is usually performed when the power is turned on.
[Setting] On the calibration screen [6 SYSTEM, Master/Cal]

Specifying coordinate systems

$MNUFRAMENUM[group] 0
BYTE RW 0 to 9

[Function] Specifies user coordinate system number
[Description] Specifies the number of the user coordinate system currently used.
0: World coordinate system
1 to 9: User coordinate system
[Setting] On the user coordinate system setting screen [6 SETUP, Frames, User Frame]

$MNUFRAME[group, 1] XYZWPR
$MNUFRAME[group, 2] XYZWPR
$MNUFRAME[group, 3] XYZWPR
$MNUFRAME[group, 4] XYZWPR
$MNUFRAME[group, 5] XYZWPR
$MNUFRAME[group, 6] XYZWPR
$MNUFRAME[group, 7] XYZWPR
$MNUFRAME[group, 8] XYZWPR
$MNUFRAME[group, 9] XYZWPR
POSITION RW XYZWPR

[Function] Specifies user coordinates system
[Description] Specifies the Cartesian coordinates in the user coordinate system. Up to nine user coordinate systems can be registered.

$MNUTOOLNUM[group] 0
BYTE RW 0 to 10

[Function] Specifies tool coordinate system number
[Description] Specifies the number of the tool coordinate system currently used.
0: Mechanical interface coordinate system
1 to 10: Tool coordinate system
[Setting] On the tool coordinate system setting screen [6 SETUP, Frames, Tool Frame]

$MNUTOOL[group, 1] XYZWPR
$MNUTOOL[group, 2] XYZWPR
$MNUTOOL[group, 3] XYZWPR
$MNUTOOL[group, 4] XYZWPR
$MNUTOOL[group, 5] XYZWPR
$MNUTOOL[group, 6] XYZWPR
$MNUTOOL[group, 7] XYZWPR
$MNUTOOL[group, 8] XYZWPR
$MNUTOOL[group, 9] XYZWPR
$MNUTOOL[group, 10] XYZWPR
POSITION RW XYZWPR

[Function] Specifies the tool coordinate system
[Description] Specify the Cartesian coordinates in the tool coordinate system. Ten tool coordinate systems can be registered.

$JOG_GROUP[group]. $JOG_FRAME XYZWPR
POSITION RW XYZWPR

[Function] Specifies the jog coordinate system
[Description] Specifies the Cartesian coordinates in the jog coordinate system.
[Setting] On the jog coordinate system setting screen [6 SETUP, Frames, Jog Frame]

Setting motors

$SCR_GRP[group]. $AXISORDER[1] 1
$SCR_GRP[group]. $AXISORDER[2] 2
$SCR_GRP[group]. $AXISORDER[3] 3
$SCR_GRP[group]. $AXISORDER[4] 4
$SCR_GRP[group]. $AXISORDER[5] 5
$SCR_GRP[group]. $AXISORDER[6] 6
$SCR_GRP[group]. $AXISORDER[7] 0
$SCR_GRP[group]. $AXISORDER[8] 0
$SCR_GRP[group]. $AXISORDER[9] 0
BYTE RW 0 to 16

[Function] Specify axis order
[Description] Specifies the order of axes by assigning the physical number of a servo motor controlled by the servo amplifier (servo register) to the logical number of a robot joint axis specified in software (Jx-axis). For instance, when $AXISORDER[1] = 2, servo motor 2 is assigned to the J1-axis. When $AXISORDER[1] = 0, no servo motor is assigned as the J1-axis.

$SCR_GRP[group]. $ROTARY_AXS[1] *
$SCR_GRP[group]. $ROTARY_AXS[2] *
$SCR_GRP[group]. $ROTARY_AXS[3] *
$SCR_GRP[group]. $ROTARY_AXS[4] *
$SCR_GRP[group]. $ROTARY_AXS[5] *
$SCR_GRP[group]. $ROTARY_AXS[6] *
$SCR_GRP[group]. $ROTARY_AXS[7] *
$SCR_GRP[group]. $ROTARY_AXS[8] *
$SCR_GRP[group]. $ROTARY_AXS[9] *
BOOLEAN RO TRUE / FALSE

[Function] Specify axis type
[Description] Specifies whether joint axes of the robot are rotational or linear.
TRUE: Rotational
FALSE: Linear

$PARAM_GROUP[group]. $MOSIGN[1] *
$PARAM_GROUP[group]. $MOSIGN[2] *
$PARAM_GROUP[group]. $MOSIGN[3] *
$PARAM_GROUP[group]. $MOSIGN[4] *
$PARAM_GROUP[group]. $MOSIGN[5] *
$PARAM_GROUP[group]. $MOSIGN[6] *
$PARAM_GROUP[group]. $MOSIGN[7] *
$PARAM_GROUP[group]. $MOSIGN[8] *
$PARAM_GROUP[group]. $MOSIGN[9] *
BOOLEAN RW PU TRUE / FALSE

[Function] Specify direction of rotation around axes
[Description] Specify whether the robot moves in the positive or negative direction when the motor rotates positively for each axis.
TRUE: The robot moves in a positive direction when the motor rotates positively.
FALSE: The robot moves in a negative direction when the motor rotates positively.

$PARAM_GROUP[group]. $ENCSCALES[1] *
$PARAM_GROUP[group]. $ENCSCALES[2] *
$PARAM_GROUP[group]. $ENCSCALES[3] *
$PARAM_GROUP[group]. $ENCSCALES[4] *
$PARAM_GROUP[group]. $ENCSCALES[5] *
$PARAM_GROUP[group]. $ENCSCALES[6] *
$PARAM_GROUP[group]. $ENCSCALES[7] *
$PARAM_GROUP[group]. $ENCSCALES[8] *
$PARAM_GROUP[group]. $ENCSCALES[9] *
REAL RW PU -10000000000 to 10000000000 (pulse/deg, pulse/mm)

[Function] Specify unit of Pulsecoder count

[Description] Specify how many pulses are required for the Pulsecoder when the robot moves around a joint axis one degree or the robot moves along a joint axis 1 mm.
Rotation axis: $ENCSCALES = 2E19 x deceleration ratio/360

$PARAM_GROUP[group]. $MOT_SPD_LIM[1] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[2] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[3] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[4] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[5] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[6] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[7] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[8] *
$PARAM_GROUP[group]. $MOT_SPD_LIM[9] *
INTEGER RW PU 0 to 100000 (rpm)

[Function] Specify maximum motor speed

[Description] Specifies the maximum speed of each servo motor for the robot for each axis. When the robot moves around or along a certain axis at a speed exceeding the maximum speed, a warning is issued. Then, the robot decelerates and moves at a speed not exceeding the maximum speed. In this case, the robot may not trace the specified path.

Override

$SHIFTOV_ENB 0
ULONG RW 0 / 1

[Function] Enables or disables shift override

[Description] The shift override function changes the feed rate override in five steps. To change the feed rate 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 [SHIFT] key, then press the override key: The feed rate override changes in the order: VFINE → FINE → 5% → 50% → 100%.

$MCR. $PROGOVERRIDE 100
INTEGER RW 0 to 100 (%)

[Function] Specifies program override

[Description] Specifies the percentage of the robot feed rate 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 of linear motion is specified by $PARAM_GROUP[group].$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.

$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 feed rate 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 feed rate override after a cold start
[Description] The feed rate override is set to this value after a cold start.

$SCR. $COORDOVRD 10
INTEGER RO 0 to 100 (%)

[Function] Specifies maximum feed rate override when the manual-feed coordinate system is changed
[Description] The feed rate 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 feed rate override when the teach pendant is enabled
[Description] The feed rate override is set to this value when the teach pendant is enabled.

$SCR. $JOGOVLIM 100
INTEGER RO 0 to 100 (%)

[Function] Specifies the maximum feed rate override during jog feed
[Description] The feed rate override is set to this value or less during jog feed.

$SCR. $RUNOVLIM 100
INTEGER RW 0 to 100 (%)

[Function] Specifies the maximum feed rate override when the program is executed
[Description] The feed rate override is set to this value or less when the program is executed.

$SCR. $FENCEOVRD 10
INTEGER RO 0 to 100 (%)

[Function] Maximum feed rate override when the safety fence is open
[Description] When the safety fence is opened (*SFSPD input is turned off), the feed rate override is set to this value or below.

$SCR. $SFJOGOVLIM 50
INTEGER RO 0 to 100 (%)

[Function] Maximum feed rate override of jog feed when the safety fence is open

[Description] If jog feed is performed while the safety fence is open, the feed rate override is set to this value or below.

$SCR. $SFRUNOVLIM 30
INTEGER RO 0 to 100 (%)

[Function] Maximum feed rate 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 feed rate override is set to this value or below.

$SCR. $RECOV_OVRD FALSE
BOOLEAN RW TRUE/FALSE

[Function] Function to restore feed rate override when the safety fence is closed

[Description] When the safety fence is closed (*SFSPD input set on), the previous feed rate 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 feed rate 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 SETUP, General]

Jog feed rate(joint feed) (deg/sec, mm/sec) =

$$\text{Maximum joint feed rate} \times \frac{\text{Each axis jog override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Jog feed rate(linear feed) (mm/sec) =

$$\text{Maximum linear feed rate} \times \frac{\text{Jog override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Jog feed rate(Circular feed) (deg/sec) =

$$\text{Maximum circular feed rate} \times \frac{\text{Orientation Jog override}}{100} \times \frac{\text{Feedrate override}}{100}$$

Each axis jog override	$SCR_GRP[g]. $JOGLIM_JNT[i] (%)
Jog override	$SCR. $JOGLIM (%)
Orientation jog override	$SCR. $JOGLIMROT (%)

Note: g is group number. i is axis number.

Operating speed (joint control motion) (deg/sec, mm/sec) =

$$\text{Maximum joint feed rate} \times \frac{\text{Programmed feedrate}}{100} \times \frac{\text{Feedrate override}}{100}$$

Operating speed (motion under path control) (mm/sec) =

$$\text{Programmed feed rate} \times \frac{\text{Feedrate override}}{100}$$

Operating speed (motion under attitude control) (deg/sec) =

$$\text{Programmed feed rate} \times \frac{\text{Feedrate override}}{100}$$

Payload specification

> **NOTE**
> Use Payload Setting screen (Motion Performance Screen) to set values of payload weight, gravity center position and inertia.

$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] Indicates the value of Inertia about each axis resulting from the 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:

$$\$AXISINERTIA[i] = \frac{payload \times (l\_max [i])^2}{g} \quad (kgf \cdot cm \cdot sec^2)$$

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 that can be achieved.
g : Gravity acceleration (= 980 [cm/sec^2])

[NOTE 1] When specifying or changing this variable, refer to the explanation of $PARAM_GROUP[].$AXIS_IM_SCL, below.

[NOTE 2] If $PARAM_GROUP[group].$SV_DMY_LNK[4] is TRUE, these system variables are automatically updated according to the present payload setting.

$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] Indicates the value of Moment about each axis resulting from the 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:

$$\$AXISMOMENT[i] = payload \times l\_max[i] \quad (kgf \cdot m)$$

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 that can be achieved.

[NOTE 1] When specifying or changing this variable, refer to the explanation of $PARAM_GROUP[].$AXIS_IM_SCL, below.

[NOTE 2] If $PARAM_GROUP[group].$SV_DMY_LNK[4] is TRUE, these system variables are automatically updated according to the present payload setting.

$PARAM_GROUP [group] . $AXIS_IM_SCL 1
SHORT RW PU 0 to 32767

[Function] Inertia and moment value adjustment scale

[Description] This scale is used to set up a number in decimal places for the inertia and moment values of each axis stated above.

[NOTE] It is usually unnecessary to re-set this variable.
Actually, the following inertia and moment values are used.

(Inertia value) = $\dfrac{\text{\$PARAM\_GROUP[group].\$AXISINERTIA[i]}}{\text{\$PARAM\_GROUP[group].\$AXIS\_IM\_SCL}}$

(Moment value) = $\dfrac{\text{\$PARAM\_GROUP[group].\$AXISMOMETN[i]}}{\text{\$PARAM\_GROUP[group].\$AXIS\_IM\_SCL}}$

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.

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 : 1 1 0 0 2 (Meaning: SERVO-002 alarm)
Alarm ID Alarm number

User alarm

$UALRM_SEV[ ] 6
BYTE RW 0 to 255

[Function] User alarm severity

[Description] Sets the user alarm severity. $UALRM_SEV[i] corresponds to the severity of user alarm[i].

0 WARN
6 STOP.L
38 STOP.G
11 ABORT.L
43 ABORT.G

The initial severity for each user alarm is 6 (STOP.L).

Jogging

$JOG_GROUP. $FINE_DIST 0.5
REAL RW 0.0 to 1.0 (mm)

[Function] Move distance for linear step jogging

[Description] Specify an amount of travel in low-speed linear step feed by Cartesian/tool manual feed. The amount of travel in very low speed step feed is one tenth of the value specified here.

$SCR . $FINE_PCNT 10
INTEGER RO 1 to 100 %

[Function] Move distance for joint or orientation step jogging

[Description] Specify an amount of travel for step feed in attitude rotation by axial manual feed or Cartesian/tool manual feed. Specify manual feed with a percentage and an override of 1%.

I/O setting

$OPWORK . $UOP_DISABLE *
BYTE RW 0 / 1

[Function] Enable/disable UOP I/O

[Description] Specify whether the peripheral equipment input signal is enabled or disabled.
If the peripheral equipment input signal is enabled when the robot is operated without any peripheral equipment connected, an alarm cannot be cleared. By disabling the signal with this setting, the alarm can be cleared. When any peripheral equipment is connected, set this variable to 0 before using that equipment.

$SCR . $RESETINVERT FALSE
BOOLEAN RW TRUE / FALSE

[Function] FAULT_RESET input signal detection.

[Description] When you set this value to "TRUE", an error is reset by rising edge of FAULT_RESET input signal. If "FALSE" is set, an error is reset by falling edge is detected.
TRUE: Check rising edge of reset input signal.
FALSE: Check falling edge of reset input signal.

$PARAM_GROUP . $PPABN_ENBL FALSE
BOOLEAN RW TRUE / FALSE

[Function] Enable/disable pressure abnormal *PPABN input

[Description] Specifies if pressure abnormal signal is detected or not. If you want to use *PPABN input, you should set this variable to TRUE.
TRUE: Enable
FALSE: Disable

$PARAM_GROUP. $BELT_ENBLE FALSE
BOOLEAN RW TRUE / FALSE

[Function] Belt rupture signal enabled/disabled

[Description] Specify whether the belt rupture signal (RI[7]) is detected.
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

[Description] On the file screen, displays the name of a file to be saved as Application.

$FILE_SYSBCK

[Description] On the file screen, displays the name of a file to be saved as System file.

Register speed specification function

$RGSPD_PREXE FALSE
BOOLEAN RO TRUE/FALSE

[Function] Advanced register speed read enabled or disabled
[Description] Specify whether an advanced read of operation statement is performed (enabled) or not (disabled) when the movement speed specified by an operation statement is held in a register.
TRUE: Advanced read enabled.
FALSE: Advanced read disabled.

> ⚠ **CAUTION**
> When an advanced register speed read is enabled with the setting indicated above, the timing at which the register value is changed is important. With some timings, a change in the register value may not be reflected in the operation speed, and the register value existing before the change may be applied to the movement. To enable advanced register speed read, some consideration is needed: The value of a register used for the movement speed during program execution should not be changed; An interlock should be provided.

Specifying an output signal of the BZAL/BLAL alarm

$BLAL_OUT.$DO_INDEX 0
INTEGER RW 0 to 256

[Description] When a non-zero number is specified, DO corresponding to that number is turned on at the occurrence of BZAL/BLAL. DO stays on until the voltage is restored by the replacement backup battery or some other means. (If a program or the I/O screen is used to turn off DO forcibly, DO is turned back on immediately.)

$BLAL_OUT.$BATALM_OR TRUE
INTEGER RW TRUE/FALSE

[Description] Specifies whether to set BATALM, a dedicated output signal, so that it has also the BZAL/BLAL function.

⚠ **CAUTION**

In case that the BATALM signal is specified to include BZAL/BLAL of Pulsecoder indicated above, BATALM is output when at least one Pulsecoder of all axis of all motion group detects BZAL/BLAL. However BATALM signal excludes Process axis (Servo torch axis) which has no motion group and Slave axis of Dual drive function.

Setup for changing jog group according to the motion group of selected program

$PROGGRP_TGL 0

INTEGER RW 0 to 2147483647

[Description] Setup changing jog group according to the motion group of selected program

1Bit(1):Setup for changing jog group according to the motion group of selected program

0:Disable (Default)

1:Enable (Jog group is changed step by step according to the motion group of selected program.)

2Bit(2):Setup for enable settings of 1Bit and 3Bit in only case T1 mode when 1Bit or 3Bit is enabled.

0: Settings of 1Bit and 3Bit is enabled regardless of the state of mode switch.

1: Settings of 1Bit and 3Bit is enabled in only T1 mode.

3Bit(4) : Setup for changing jog group automatically when program is selected

0: Disable (Default)

1: Enable (Jog group is changed to motion group of selected program when program is selected in select menu.)

NOTE

n of nBit means low n digit in binary digit. For example, in case 1Bit is 1 and 3Bit is 1 others are 0, “00000101” in binary digit represents “5” in decimal digit. So input “5” to $PROGGRO_TGL.

Default setting for motion group

$DSBL_GPMSK 0

INTEGER RW 0 to 255

[Description] The specified motion group of the program is disable when the program is created. The motion group is specified as bit (1-8) of this variable. For example, motion groups 1 and 3 are disable in case $DSBL_GPMSK is 5, the default motion group of the program will be setup to [*,1,*,1,*,*, *,*] in case system has 4 group when program is created.

NOTE

n of nBit means low n digit in binary digit. For example, in case 1Bit is 1 and 3Bit is 1 others are 0, “00000101” in binary digit represents “5” in decimal digit. So input “5” to $DSBL_GPMSK.

Motion command Information

$CMD_INFO [group] . $TRQ_CMD [1] ∗
$CMD_INFO [group] . $TRQ_CMD [2] ∗
$CMD_INFO [group] . $TRQ_CMD [3] ∗
$CMD_INFO [group] . $TRQ_CMD [4] ∗
$CMD_INFO [group] . $TRQ_CMD [5] ∗
$CMD_INFO [group] . $TRQ_CMD [6] ∗
$CMD_INFO [group] . $TRQ_CMD [7] ∗
$CMD_INFO [group] . $TRQ_CMD [8] ∗
$CMD_INFO [group] . $TRQ_CMD [9] ∗

REAL RW -1000000000.0 ～ 1000000000.0 (Ap)

［Function］ Torque command.

［Description］ Command value of current for each motor to output the torque.

$CMD_INFO [group] . $JNT_POS [1] ∗
$CMD_INFO [group] . $JNT_POS [2] ∗
$CMD_INFO [group] . $JNT_POS [3] ∗
$CMD_INFO [group] . $JNT_POS [4] ∗
$CMD_INFO [group] . $JNT_POS [5] ∗
$CMD_INFO [group] . $JNT_POS [6] ∗
$CMD_INFO [group] . $JNT_POS [7] ∗
$CMD_INFO [group] . $JNT_POS [8] ∗
$CMD_INFO [group] . $JNT_POS [9] ∗

REAL RW -1000000000.0 ～ 1000000000.0 (deg or mm)

［Function］ Axis position command.

［Description］ Command position of each axis. The unit is [deg] for rotational axis, [mm] for linear axis.

$CMD_INFO [group] . $CART_POS [1] ∗
$CMD_INFO [group] . $CART_POS [2] ∗
$CMD_INFO [group] . $CART_POS [3] ∗
$CMD_INFO [group] . $CART_POS [4] ∗
$CMD_INFO [group] . $CART_POS [5] ∗
$CMD_INFO [group] . $CART_POS [6] ∗
$CMD_INFO [group] . $CART_POS [7] ∗
$CMD_INFO [group] . $CART_POS [8] ∗
$CMD_INFO [group] . $CART_POS [9] ∗

REAL RW -1000000000.0 ～ 1000000000.0 (deg or mm)

［Function］ Cartesian position command (World coordinate system).

［Description］ Command position of TCP with respect to world coordinate system. Represents X, Y, Z, W, P, R in order from array [1]. The unit is [mm] for X, Y, Z, [deg] for W, P, R.

$CMD_INFO [group] . $CART_POS_UF [1] ∗
$CMD_INFO [group] . $CART_POS_UF [2] ∗
$CMD_INFO [group] . $CART_POS_UF [3] ∗
$CMD_INFO [group] . $CART_POS_UF [4] ∗
$CMD_INFO [group] . $CART_POS_UF [5] ∗
$CMD_INFO [group] . $CART_POS_UF [6] ∗
$CMD_INFO [group] . $CART_POS_UF [7] ∗
$CMD_INFO [group] . $CART_POS_UF [8] ∗
$CMD_INFO [group] . $CART_POS_UF [9] ∗
REAL RW -1000000000.0 ～ 1000000000.0 (deg or mm)

［Function］ Cartesian position command (User coordinate system).

［Description］ Command position of TCP with respect to user coordinate system. Represents X, Y, Z, W, P, R in order from array [1]. The unit is [mm] for X, Y, Z, [deg] for W, P, R.

NOTE
1 $CMD_INFO is updated every about 100 msec.
2 $CMD_INFO is available in the version 7DC3/44 or later and 7DF1/15 or later.

Servo Information

$SV_INFO [group] . $Q_CURRENT [1] ∗
$SV_INFO [group] . $Q_CURRENT [2] ∗
$SV_INFO [group] . $Q_CURRENT [3] ∗
$SV_INFO [group] . $Q_CURRENT [4] ∗
$SV_INFO [group] . $Q_CURRENT [5] ∗
$SV_INFO [group] . $Q_CURRENT [6] ∗
$SV_INFO [group] . $Q_CURRENT [7] ∗
$SV_INFO [group] . $Q_CURRENT [8] ∗
$SV_INFO [group] . $Q_CURRENT [9] ∗
REAL RW -1000000000.0 ～ 1000000000.0 (Ap)

［Function］ Q-phase current command.

［Description］ The effective component of current Feedback.

$SV_INFO [group] . $AXIS_POS [1] ∗
$SV_INFO [group] . $AXIS_POS [2] ∗
$SV_INFO [group] . $AXIS_POS [3] ∗
$SV_INFO [group] . $AXIS_POS [4] ∗
$SV_INFO [group] . $AXIS_POS [5] ∗
$SV_INFO [group] . $AXIS_POS [6] ∗
$SV_INFO [group] . $AXIS_POS [7] ∗
$SV_INFO [group] . $AXIS_POS [8] ∗
$SV_INFO [group] . $AXIS_POS [9] ∗
REAL RW -1000000000.0 ～ 1000000000.0 (deg or mm)

［Function］ Axis position.

［Description］ Position of each axis. The unit is [deg] for rotational axis, [mm] for linear axis.

$SV_INFO [group] . $CART_POS [1] ∗
$SV_INFO [group] . $CART_POS [2] ∗
$SV_INFO [group] . $CART_POS [3] ∗
$SV_INFO [group] . $CART_POS [4] ∗
$SV_INFO [group] . $CART_POS [5] ∗
$SV_INFO [group] . $CART_POS [6] ∗
REAL RW -1000000000.0 ～ 1000000000.0 (deg or mm)

［Function］ Cartesian position (World coordinate system).

［Description］ TCP position with respect to world coordinate system. Represents X, Y, Z, W, P, R in order from array [1]. The unit is [mm] for X, Y, Z, [deg] for W, P, R.

$SV_INFO [group] . $CART_POS_UF [1] ∗
$SV_INFO [group] . $CART_POS_UF [2] ∗
$SV_INFO [group] . $CART_POS_UF [3] ∗
$SV_INFO [group] . $CART_POS_UF [4] ∗
$SV_INFO [group] . $CART_POS_UF [5] ∗
$SV_INFO [group] . $CART_POS_UF [6] ∗
REAL RW -1000000000.0 ～ 1000000000.0 (deg or mm)

［Function］ Cartesian position (User coordinate system).

［Description］ TCP position with respect to user coordinate system. Represents X, Y, Z, W, P, R in order from array [1]. The unit is [mm] for X, Y, Z, [deg] for W, P, R.

NOTE

1. The axis position may not always match the command position since the actual TCP position is controlled to match the command position.
2. The Cartesian position is a theoretical position calculated from the machine pulse of each axis, and does not accurately represent the position in the real space.
3. $Q_CURRENT and $AXIS_POS are available in the version 7DC2/44 or later, 7DC3/17 or later and 7DF1/01 or later.
4. $CART_POS and $CART_POS_UF are available in the version 7DC3/44 or later and 7DF1/15 or later.

System timer

$SYSTEM_TIME[group].PWR_TOT 0
INTEGER RO 0 to 2147483647

[Function] On Power time.

[Description] Time during which the power to the controller is on (Unit: minute).

$SYSTEM_TIME[group].SRV_TOT 0
INTEGER RO 0 to 2147483647

[Function] On Power time.

[Description] Time during which the system is ready for operation (servo on) after the release of an alarm (Unit: minute).

$SYSTEM_TIME[group].RUN_TOT 0
INTEGER RO 0 to 2147483647

[Function] Running time.

[Description] Program execution time. The halt period is not included (Unit: minute).

$SYSTEM_TIME[group].WIT_TOT 0
INTEGER RO 0 to 2147483647

[Function] Waiting time.
[Description] Time required to execute a WAIT instruction (Unit: minute).

$SYSTEM_TIME[group].SHM_TOT 0
INTEGER RO 0 to 2147483647

[Function] Servo hour meter time.
[Description] Time during which servo brake is released (Unit: minute).

System ready

$PWRUP_DELAY.$SY_READY FALSE
BOOLEAN RO TRUE/FALSE

[Function] System ready.
[Description] FALSE means that system is NOT ready(during Start-up). TRUE means that system is ready.

D SAVING RESEARCH DATA

Overview

This is the procedure manual of saving necessary data for investigating some abnormal status or issues. You can save the research data into the memory card (*) to execute this manual in order. Please send the saving data to FANUC.

(*) In the case of R-30iB Mate controller, please replace the memory card to USB memory.

Needed Equipment

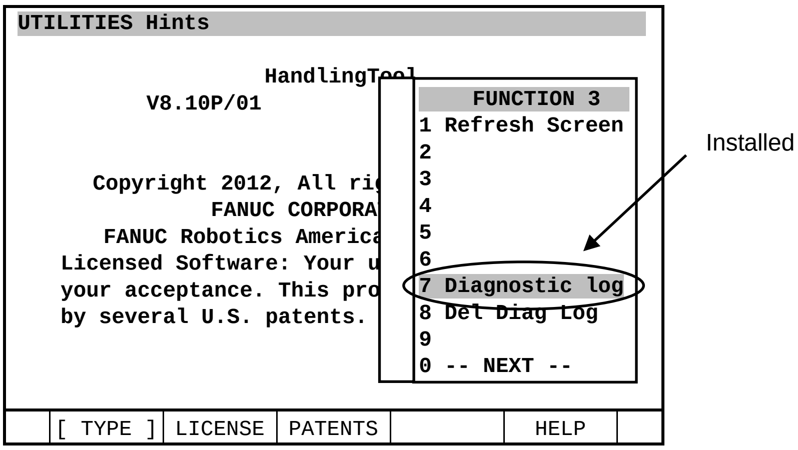
One flash ATA memory card over 256MB (it just calls memory card in this manual):
Usually research data can be saved when memory card has over 128MB vacant space. However in the case of a lot of files are existed in it, data cannot be saved normally. Then we recommend to use empty memory card.

Method of Saving Research Data

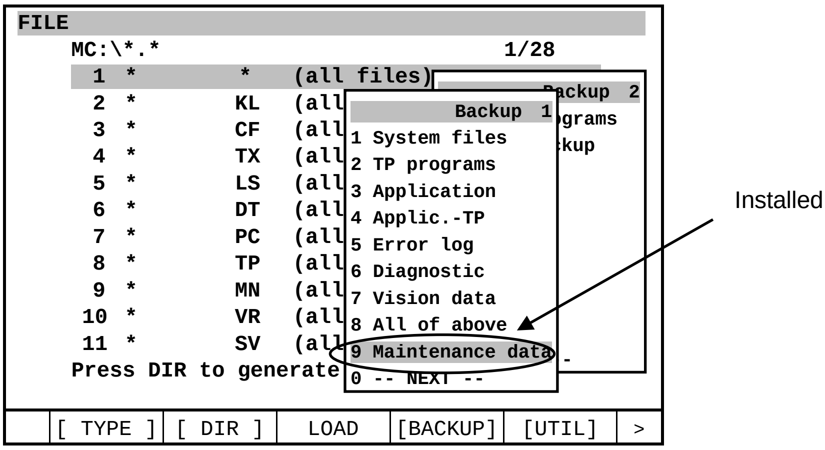
NOTE
The figure in this manual may be different from your system.

You can save its data by doing next page procedure in order. Procedure is different from if following functions are installed or not. If yes, like following screen is displayed.

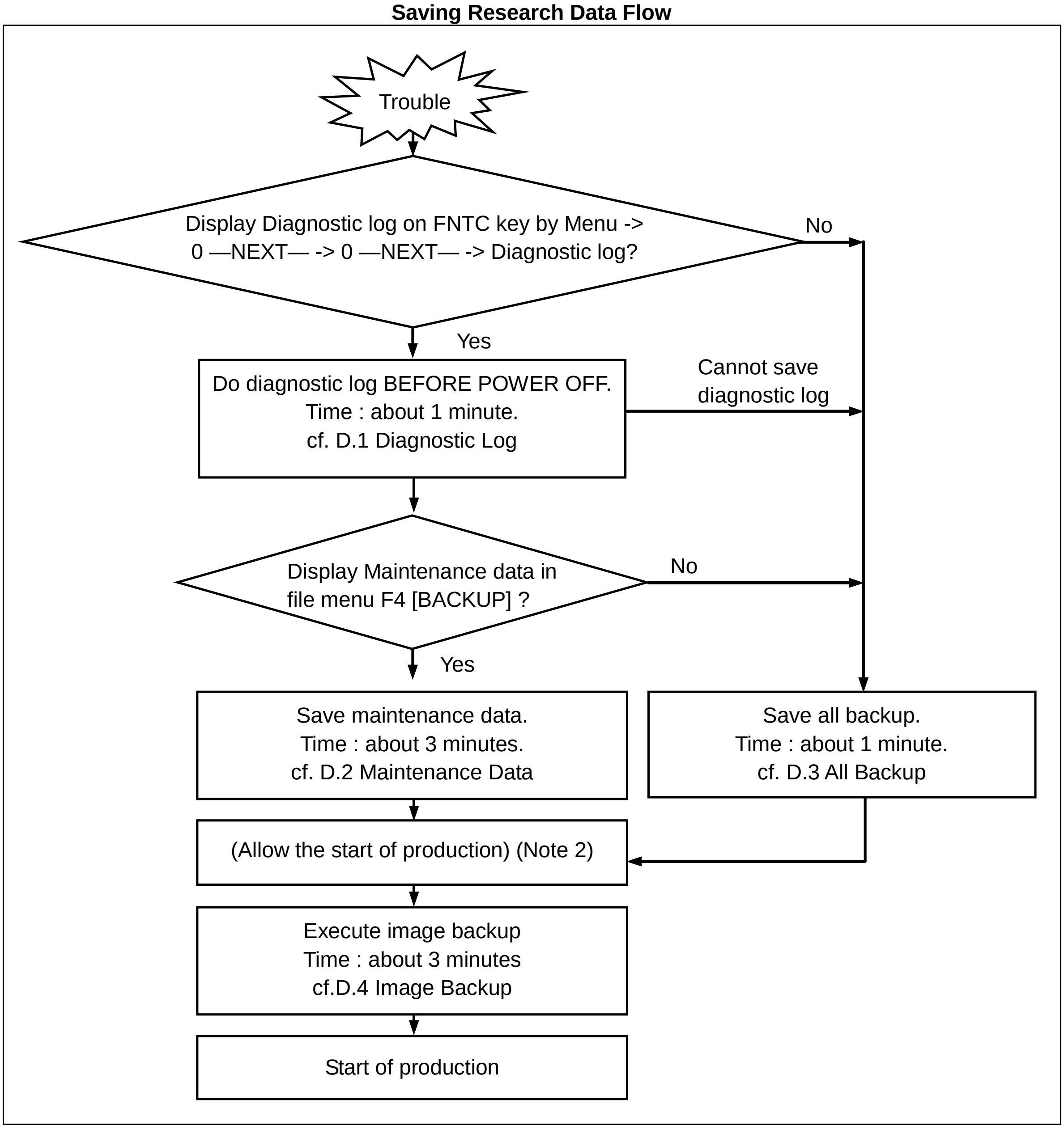
Diagnostic Log (Select [FCTN] key on TP → 0--NEXT -- → 0--NEXT-- → 7 Diagnostic log) :



Maintenance Data (Select Menu – File – F4, [BACKUP] – Maintenance data) :



Saving Research Data Flow



NOTE

1. Time may differ largely in your system.
2. If you have time after doing maintenance data or all backup, please do image backup before starting production.

D.1 DIAGNOSTIC LOG

⚠WARNING

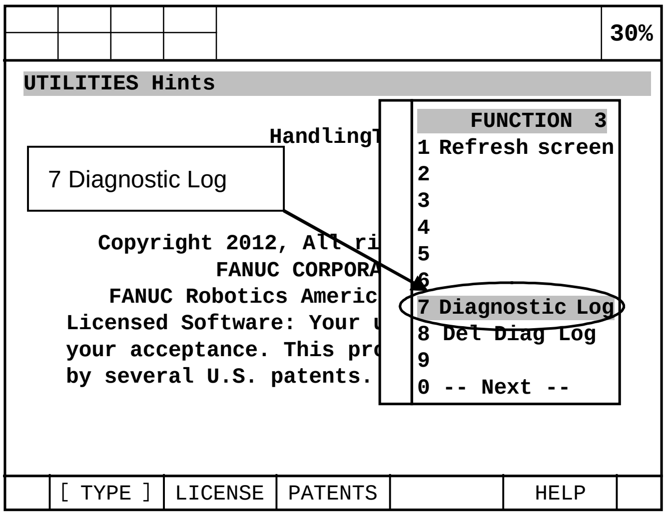
If you execute this function, verify that the status of robot is STOP. In some situations robot becomes jerky motion while saving log.

NOTE

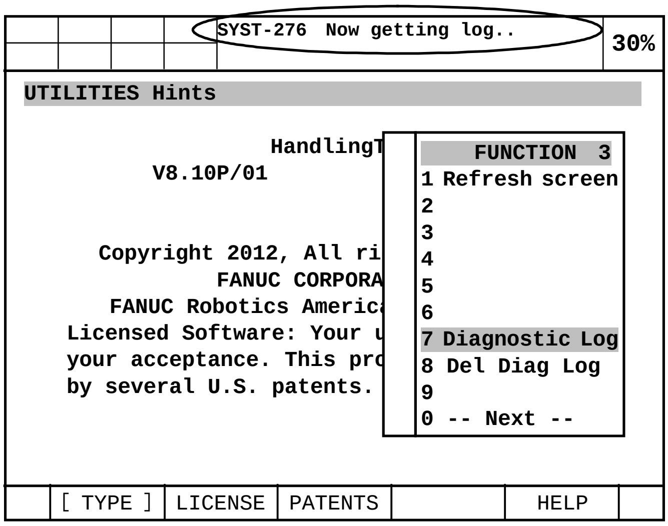
- Do below operations BEFORE POWER OFF at the time of a trouble occurs. If you power off, available data for investigating is lost.
- Save diagnostic log just 1 time immediately after problem occurs. If the diagnostic log is saved more than once continuously for same problem, the data at the time when the problem has occurred is deleted.

Data is saved into FROM, therefore you do not have to insert the external memory such as memory card or USB.

1. Select [FCTN] key -> 0--NEXT-- -> 0--NEXT-- -> 7 Diagnostic log.
2. "Diagnostic log" is displayed on 3rd page in FCTN menu.

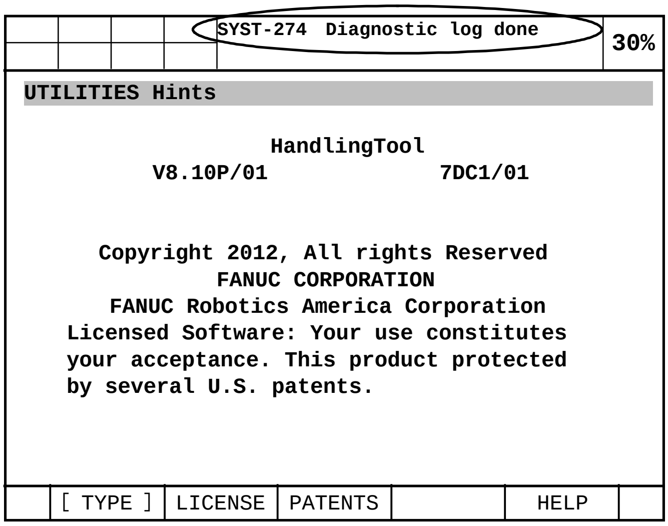


3. Set the cursor onto "Diagnostic log" and press the [ENTER] key. SYST – 276 Now getting log… is posted.



4 It takes about 1 minute to save.

5 SYST – 274 Diagnostic log done is posted after done saving. If this warning does not appear, you can confirm by the disappearance of FUNCTION menu.



NOTE

1. If you see “SYST – 275 Diagnostic log failed”, please save all backup before power off, do image backup and send its data to FANUC. At this time please notify the procedure of duplication its problem too.
2. Depending on the condition of occurring problem, this function is not available. For example, in the case TP is frozen and any key operation are ignored. In that case, turn the power off and on, save the all backup and the image backup, and send the data and the method of reproducing the problem to fanuc.
3. If you execute this function again, data is overwritten. Diagnostic log is saved only 1 time. The old data will be deleted after the second time. When executing Diagnostic log two or more times, it is necessary to save the image backup every time.

If “SYST-273 Lack of FRA for saving” occurs,

It is possible to output the data of diagnostic log to the external storage device. Refer to the Subsection “9.15.4 Output to External Device”.

If the message for the lack of FROM is displayed after saving diagnostic log,

Execute image backup, then delete the saved diagnostic log by the function menu.

```
FROM has low free space.
Execute image backup and
Delete diagnostic log.


          [ OK ]
```

Select FCTN -> 0--NEXT-- -> 0--NEXT-- -> 8 Del Diag Log.
For detail about deletion, refer to the Subsection “9.15.5 Delete the Data saved in FROM”.

D.2 MAINTENANCE DATA

> **NOTE**
> Saving time of maintenance data depends on the system. It takes about 3 minutes as a rough guide. In the case that a lot of TPP program or many directories for auto backup are existed, saving time get longer.

1. Insert the memory card to the controller.
2. Press [MENU] key – 7 FILE.
3. If MC: is not displayed in the upper left portion on the screen, select F5, [UTIL] – Set Device – Mem Card(MC:).
4. Select – F4, [BACKUP] – Maintenance data.

```
FILE
    MC:\*.*                                  1/28
     1  *        *    (all files)
     2  *       KL    (al             Backup 1    Backup 2
     3  *       CF    (al 1 System files          ograms
     4  *       TX    (al 2 TP programs           ckup
     5  *       LS    (al 3 Application
     6  *       DT    (al 4 Applic.-TP
     7  *       PC    (al 5 Error log
     8  *       TP    (al 6 Diagnostic
     9  *       MN    (al 7 Vision data
    10  *       VR    (al 8 All of above
    11  *       SV    (al 9 Maintenance data
    Press DIR to generate 0 -- NEXT --

  [ TYPE ] [ DIR ]  LOAD  [BACKUP] [UTIL]  >
```

5 Select F4, YES after displaying the confirm message.

```
FILE
    MC:\*.*                            1/28
      1  *        *   (all files)
      2  *        KL  (all KAREL source)
      3  *        CF  (all command files)
      4  *        TX  (all text files)
      5  *        LS  (all KAREL listings)
      6  *        DT  (all KAREL data files)
      7  *        PC  (all KAREL p-code)
      8  *        TP  (all TP programs)
      9  *        MN  (all MN programs)
     10  *        VR  (all variable files)
     11  *        SV  (all system files)
    Save maintenance data. OK?

                          YES      NO
```

6 In above sample case, maintenance data is saved into MC:\MNT_DATA\. It is saved into MNT_DATA\ under the current directory. MNT_DATA is newly created.

D.3 All BACKUP

1 Insert the memory card to the controller.
2 Press [MENU] key , then select 7 FILE.
3 For prevention of miss operation that deletes the data existed in the card, new directory is created. Select F5, [UTIL] – Make DIR.

```
FILE
    MC:\*.*                            1/28
      1  *        *   (all files)
      2  *        KL  (all KAREL source)
      3  *        CF  (all command files)
      4  *        TX  (all text files)
      5  *        LS  (all KAREL listings)
      6  *        DT  (all KAREL data files)
      7  *        PC  (all KAREL p   UTIL    1
      8  *        TP  (all TP prog 1 Set Device
      9  *        MN  (all MN prog 2 Format
     10  *        VR  (all variabl 3 Format FAT32
     11  *        SV  (all system  4 Make DIR
    Press DIR to generate director

 [ TYPE ] [ DIR ]  LOAD  [BACKUP] [UTIL ]  >
```

4 Input the directory name. For convenience of explanation, directory name is set “Year, Month, Day, Number”. In this sample 090810_1 is used.

```
FILE
    MC:\*.*                              1/28
     1  *        *    (all files)
     2  *       KL    (all KAREL source)
     3  *       CF    (all command files)
     4  *       TX    (all text files)
     5  *       LS    (all KAREL listings)
     6  *       DT    (all KAREL data files)
     7  *       PC    (all KAREL p-  Alpha input  1
     8  *       TP    (all TP progr    Words
     9  *       MN    (all MN progr    Upper Case
    10  *       VR    (all variable    Lower case
    11  *       SV    (all system f    Options
    Directory name: 090810_1

   ABCDEF   GHIJKL   MNOPQR   STUVWX   YZ_@*.
```

5 Select F4, [BACKUP] – All of above.

```
FILE
    MC:\090810_1\*.*                     1/29
     1  ..    (Up one level)           Backup  2
     2  *        *    (al        Backup  1  ograms
     3  *       KL    (al 1 System files    ckup
     4  *       CF    (al 2 TP programs
     5  *       TX    (al 3 Application
     6  *       LS    (al 4 Applic.-TP
     7  *       DT    (al 5 Error log
     8  *       PC    (al 6 Diagnostic
     9  *       TP    (al 7 Vision data
    10  *       MN    (al 8 All of above
    11  *       VR    (al 9 Maintenance data --
    Press DIR to generate 0 -- NEXT --

   [ TYPE ] [ DIR ]   LOAD   [BACKUP] [UTIL ]  >
```

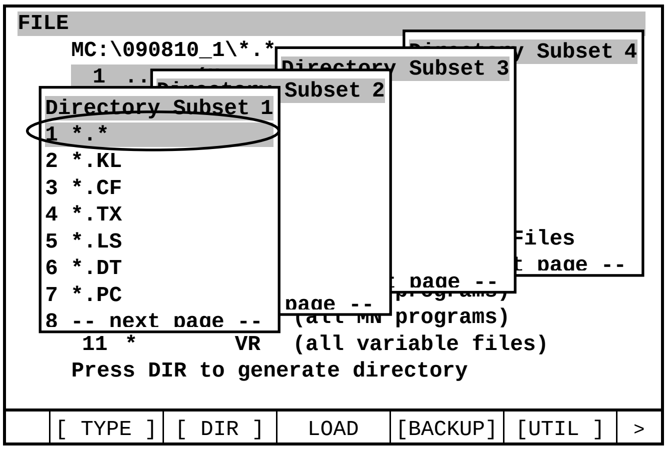
6 Select F4, YES after confirm message for deleting is displayed.

```
FILE
    MC:\090810_1\*.*                     1/29
     1  ..    (Up one level)      <DIR>
     2  *        *    (all files)
     3  *       KL    (all KAREL source)
     4  *       CF    (all command files)
     5  *       TX    (all text files)
     6  *       LS    (all KAREL listings)
     7  *       DT    (all KAREL data files)
     8  *       PC    (all KAREL p-code)
     9  *       TP    (all TP programs)
    10  *       MN    (all MN programs)
    11  *       VR    (all variable files)
    Delete MC:\090810_1\ before backup files

                                    YES      NO
```

7 Select F4, YES after confirm message for saving is displayed.

```
FILE
    MC:\090810_1\*.*                  1/29
      1  ..    (Up one level)      <DIR>
      2  *        *    (all files)
      3  *        KL   (all KAREL source)
      4  *        CF   (all command files)
      5  *        TX   (all text files)
      6  *        LS   (all KAREL listings)
      7  *        DT   (all KAREL data files)
      8  *        PC   (all KAREL p-code)
      9  *        TP   (all TP programs)
     10  *        MN   (all MN programs)
     11  *        VR   (all variable files)
    Delete MC:\090810_1\ and backup all file

          |        |        |  YES   |  NO   |
```

8 All backup is saved.
9 Select F2, [DIR] - *.* after done saving.



10 Verify that all backup is saved normally. (If several files are displayed, it is judged that it is saved.)

```
FILE
    MC:\090810_1\*.*                  1/200
      1  ..    (Up one level)      <DIR>
      2  -BCKED8-                   TP        74
      3  -BCKED9-                   TP        74
      4  -BCKEDT-                   TP        74
      5  AMBERLED                   GIF     1046
      6  ATERRJOB                   VR        46
      7  BACKDATE                   DT       274
      8  C1                         PC       587
      9  C1                         VR        40
     10  C10                        PC       589
     11  c10                        VR        41

  [ TYPE ] [ DIR ]   LOAD   [BACKUP] [UTIL]   >
```

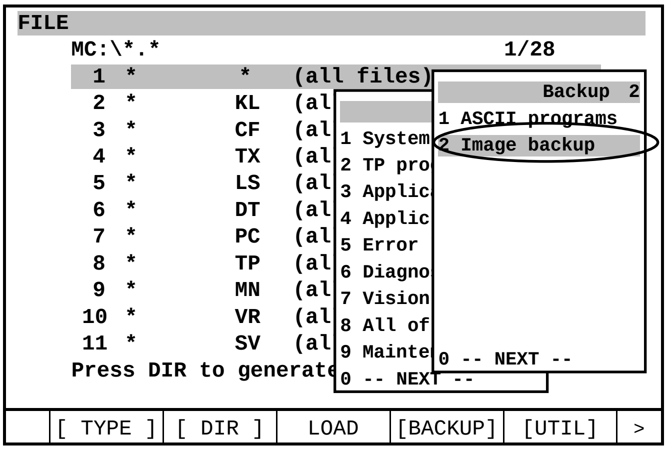
See “Batched save” in the section “FILE INPUT/ OUTPUT”.

D.4 IMAGE BACKUP

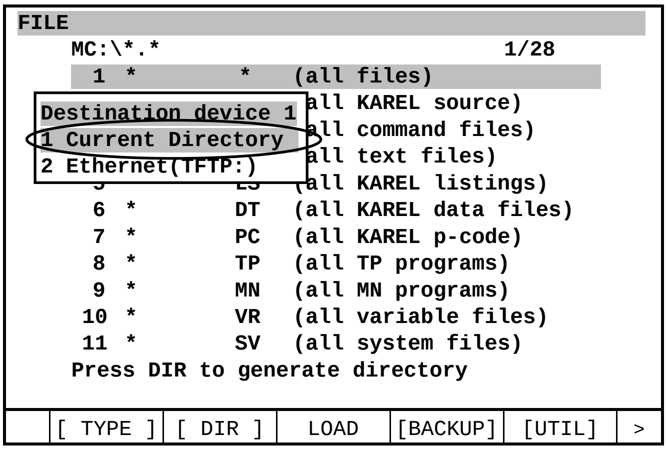
Please execute image backup after doing diagnostic log. Image backup allows the diagnostic log data to save external device such as memory card.

> ⚠CAUTION
> Executing image backup, cycle power is executed automatically. Please verify whether cycle power is done or not.

1 Insert the memory card to the controller.
2 Set the teach pendant enable switch to ON.
3 Press [MENU] key, then select 7 FILE.
4 Press F5, [UTIL], then select “Set Device” and “Mem Card (MC:)”.
5 Press F4, [BACKUP], then select “Image backup”.



6 The menu for Destination Device is displayed. Select “Current Directory”.



7 If *.IMG files already exist in the memory card, the following confirmation message is displayed. Press F4, YES to delete the existing files.

```
FILE
    MC:\*.*                           1/28
     1  *        *   (all files)
     2  *       KL   (all KAREL source)
     3  *       CF   (all command files)
     4  *       TX   (all text files)
     5  *       LS   (all KAREL listings)
     6  *       DT   (all KAREL data files)
     7  *       PC   (all KAREL p-code)
     8  *       TP   (all TP programs)
     9  *       MN   (all MN programs)
    10  *       VR   (all variable files)
    11  *       SV   (all system files)
    Remove existing IMG files ?

                          YES      NO
```

8 Select F4, YES after confirm message for cycle power is displayed.

```
FILE
    MC:\*.*                           1/28
     1  *        *   (all files)
     2  *       KL   (all KAREL source)
     3  *       CF   (all command files)
     4  *       TX   (all text files)
     5  *       LS   (all KAREL listings)
     6  *       DT   (all KAREL data files)
     7  *       PC   (all KAREL p-code)
     8  *       TP   (all TP programs)
     9  *       MN   (all MN programs)
    10  *       VR   (all variable files)
    11  *       SV   (all system files)
    Cycle power?

                          OK       CANCEL
```

9 In R-30*i*B controller, cycle power is done automatically and execute image backup. In R-30*i*B Mate controller, please cycle power. Please do not push any key until done the image backup.

10 “Image backup completed successfully.” is displayed after done the image backup. Select F4, OK.

```
FILE
    MC:\*.*                           1/28









    Image backup completed successfully.

                          OK
```

11 Data are saved like FROM**.IMG, SRAM**.IMG (** is number) in the memory card. For example, FROM32MB, SRAM2MB system is saved 34 files like FROM00.IMG, FROM01.IMG, …, FROM31.IMG, SRAM00.IMG, SRAM01.IMG.

12 Select Menu – 7 FILE – F2, [DIR] - *.*. Verify that FROM**.IMG, SRAM**.IMG are saved.

See the Subsection Image Backup Function under the Section FILE INPUT/OUTPUT.

INDEX

<T>

REVISION RECORD

Edition	Date	Contents
09	Jun, 2020	• Addition of R-30*i*B Mini Plus. • 15 OPERATIONS TABLET TEACH PENDANT has been added.
08	Nov.,2019	• Specification has been modified in some functions. • Correction of errors.
07	May.,2018	• Addition of R-30*i*B Compact Plus. • 13 OPERATIONS WITHOUT TEACH PEANDANT and 14 OPERATION WITHOUT MODE SWITCH have been added.
06	Nov.,2017	• Addition of R-30*i*B Plus and R-30*i*B Mate Plus. • 12 *i*HMI has been added. • Correction of errors.
05	Sep.,2015	Correction of errors.
04	Aug.,2013	Addition of new software series 7DC2 (V8.20).
03	Mar.,2012	Addition of R-30*i*B Mate.
02	Nov.,2012	• The contents of 9.9 LOAD SETTING have been moved to Chapter 3 and 4. • 9.9 COLLISION DETECTION FOR AUXILIARY AXIS has been added. • Specification has been modified in some functions. • Some errors in writing have been corrected.
01	Aug., 2012	

B-83284EN/09

* B - 8 3 2 8 4 E N / 0 9 *