FANUC Robot series

R-30*i*B/ R-30*i*B Mate CONTROLLER

Optional Function

OPERATOR'S MANUAL

B-83284EN-2/05

• 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 describes the precautions which must be observed to ensure the safe use of the robot.
Before attempting to use the robot, be sure to read this chapter thoroughly.

Before using the functions related to robot operation, read the relevant operator's manual to become familiar with those functions.

For the safety of the operator and the system, follow all safety precautions when operating a robot and its peripheral devices installed in a work cell.
In addition, refer to the “FANUC Robot SAFETY HANDBOOK (**B-80687EN**)”.

1 DEFINITON OF USER

The user can be classified as follows.

Operator:
- Turns the robot controller power ON/OFF
- Starts the robot program with operator’s panel

Programmer:
- Operates the robot
- Teaches the robot inside the safety fence

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

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

During the operation, programming, and maintenance of your robotic system, programmer, teaching operator and maintenance engineer must operate with circumspection by using following safety precautions.

- Adequate clothes for the operation
- Safety shoes
- A 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 or "**WARNING**" or "**CAUTION**" according to its severity. Supplementary information is indicated by "**NOTE**". Please read each "**WARNING**", "**CAUTION**" and "**NOTE**" before using the robots.

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.

3 USER SAFETY

User safety is the primary safety consideration. As it is very dangerous to enter the operating-area of the robot during its automatic operation, adequate safety precautions must be observed.

The following lists the general safety precautions. Careful consideration must be made to ensure user safety.

(1) We obligate the User to take a FANUC training courses.

FANUC provides various training courses. Contact your local FANUC representative for details.

(2) Even when the robot is stationary, it is possible that the robot is still in a ready to move state, and is waiting for a signal. In this state, the robot is regarded as still in motion. To ensure user safety, provide the system with an alarm to indicate visually or aurally that the robot is in motion.

(3) Install a safety fence with a gate so that no user can enter the safety fence inside without passing through the gate. Install an interlocking device, a safety plug, and so forth in the safety gate so that the robot is stopped as the safety gate is opened.

The controller is designed to receive this interlocking signal of the door switch. When the safety fence is opened and this signal received, the controller stops the robot (Please refer to **"STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS** for detail of stop type). For connection, refer to below **Fig.3 (b).**

(4) Provide the peripheral devices with appropriate grounding (Class A, Class B, Class C, and Class D).

(5) Recommend to install the peripheral device outside of the operating space.

(6) Draw an outline on the floor, clearly indicating the range of the robot motion, including the tools such as a hand.

(7) Install a mat switch or photoelectric switch on the floor with an interlock to a visual or aural alarm that stops the robot when a user enters the operating space.

(8) If necessary, install a safety lock so that no one except the user in charge can turn the power on the robot.

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

(9) When adjusting each peripheral device independently, make sure to turn the power off the robot.

(10) Operators must take the gloves off while manipulating the operator’s panel or teach pendant. Operation with gloved fingers may cause an operation error.

(11) Programs, system variables, and other information can be saved on memory card or USB memories. Be sure to save the data periodically in case the data is lost in an accident.

(12) The robot must be transported and installed by accurate procedure recommended by FANUC. Wrong transportation or installation may cause the robot to fall, resulting in severe injury to workers.

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

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

(15) Do not operate the robot under the following conditions. Otherwise, the robot and peripheral equipment can be adversely affected, or workers can be severely injured.
- Flammable
- Explosive
- Massive dose of Radiation
- Under water, high (heavy) Humidity
- Transport human or animals
- Stepladder (climb or hang down)
- Outdoor

(16) When connecting the peripheral devices related to stop(safety fence etc.) and each signal (external emergency , fence etc.) of robot, be sure to confirm the stop movement and do not take the wrong connection.

(17) In preparing the trestle, please secure the maintenance engineer safety at high place in reference to Fig. 3 (c). Design with the Scaffolding and Safety-belt with circumspection.

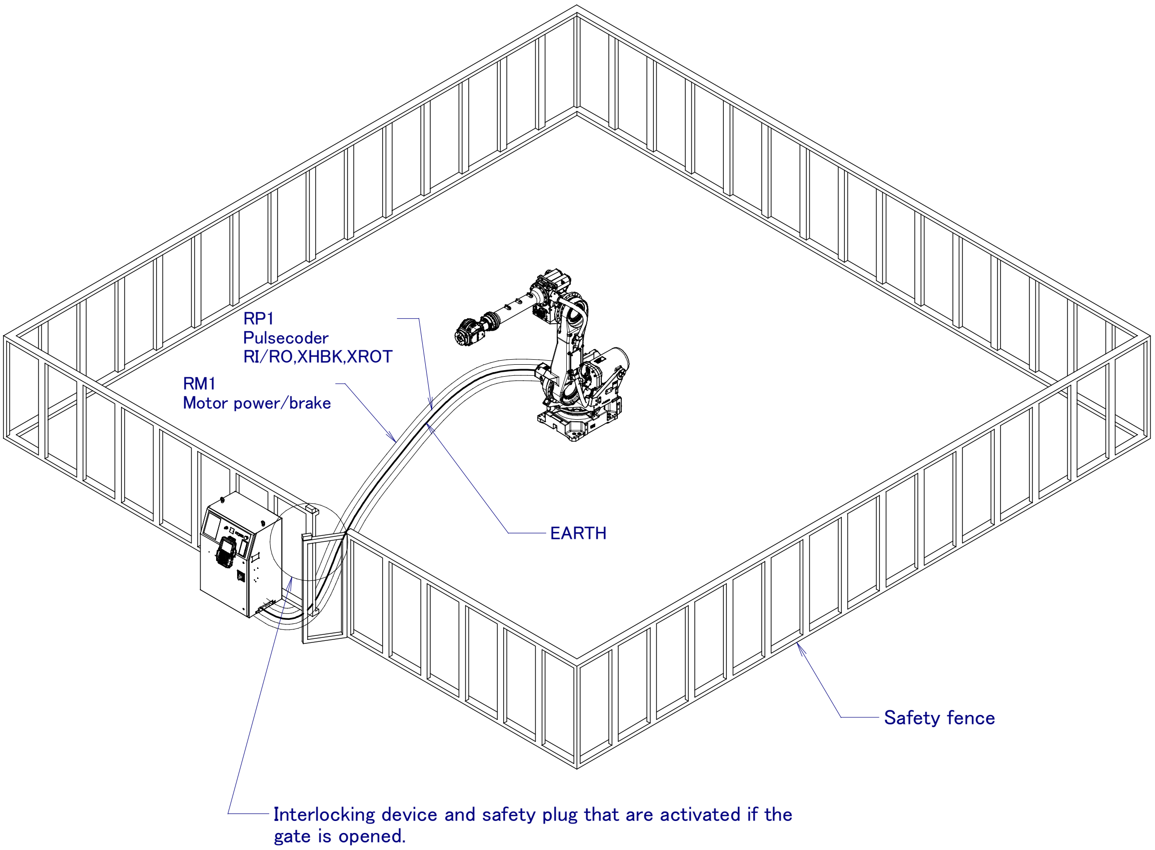
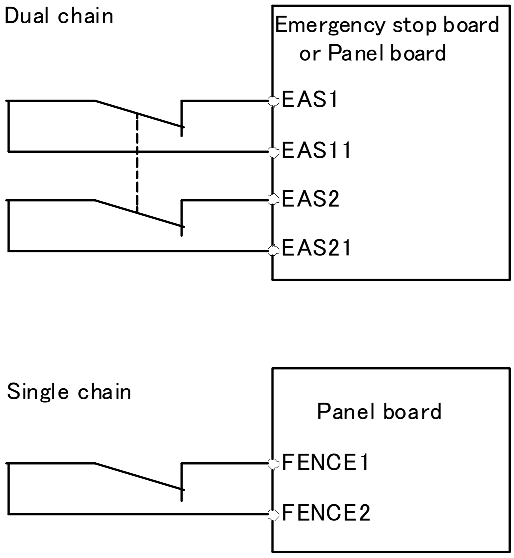

RP1
Pulsecoder
RI/RO,XHBK,XROT
RM1
Motor power/brake
EARTH
Safety fence
Interlocking device and safety plug that are activated if the gate is opened.

Fig. 3 (a) Safety fence and safety gate

> ⚠**WARNING**
> When you close a fence, please confirm that there is not a person from all directions of the robot.



(Note)

For the R-30*i*B, the R-30*i*B Mate
Terminals EAS1,EAS11,EAS2,EAS21 are provided on the emergency stop board.

Refer to the ELECTRICAL CONNCETIONS Chapter of CONNECTION of
R-30*i*B controller maintenance manual (B-83195EN) or
R-30*i*B Mate controller maintenance manual (B-83525EN)
for details.

Fig. 3 (b) Connection diagram for the signal of safety fence

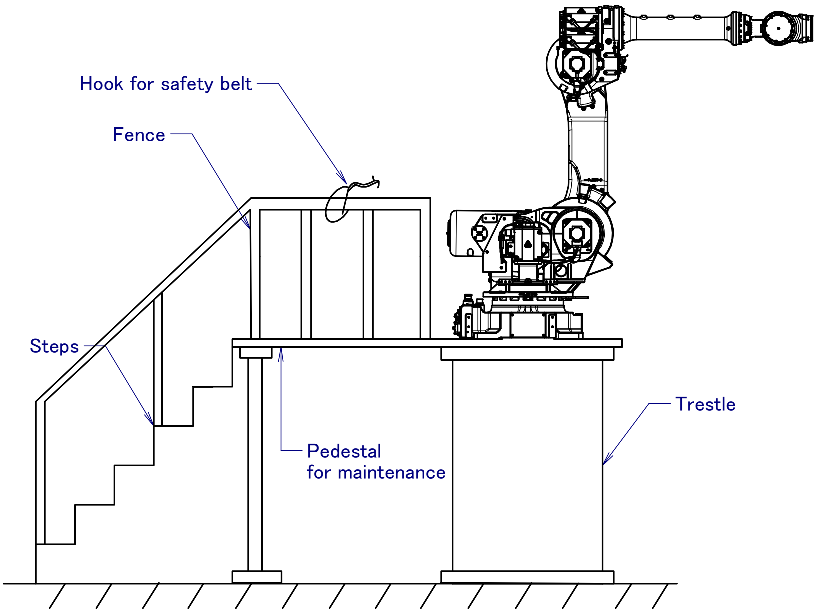


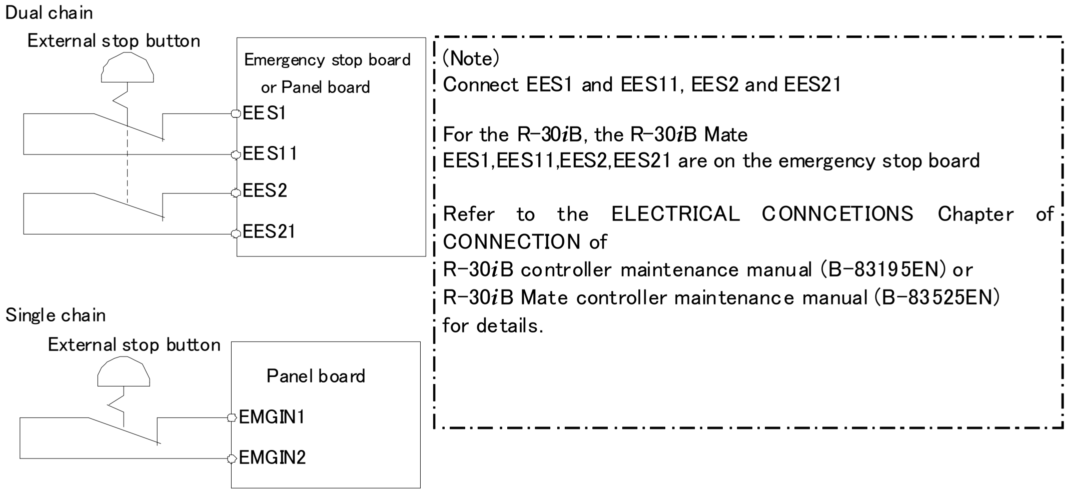
Fig. 3 (c) Pedestal for maintenance

3.1 OPERATOR SAFETY

The operator is a person who operates the robot system. In this sense, a worker who operates the teach pendant is also an operator. Operator cannot work inside the safety fence.

(1) If you don't need to operate the robot, turn the power off the robot controller, or press the "EMERGENCY STOP" button, and then proceed your work.
(2) Operate the robot system outside of the robot operating space.
(3) Install a safety fence with a safety gate to prevent any worker other than the operator from entering the dangerous area unexpectedly and the worker from entering a hazardous area.
(4) Install one or more necessary quantity of EMERGENCY STOP button(s) within the operator's reach in appropriate location(s) based on the system layout.

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



(Note)
Connect EES1 and EES11, EES2 and EES21

For the R-30*i*B, the R-30*i*B Mate
EES1,EES11,EES2,EES21 are on the emergency stop board

Refer to the ELECTRICAL CONNCETIONS Chapter of CONNECTION of
R-30*i*B controller maintenance manual (B-83195EN) or
R-30*i*B Mate controller maintenance manual (B-83525EN)
for details.

Fig. 3.1 Connection diagram for external emergency stop button

3.2 SAFETY OF THE PROGRAMMER

While teaching the robot, the operator must enter the operating space of the robot. Please ensure the safety of programmer.

(1) Unless it is specifically necessary to enter the robot operating space, carry out all tasks outside the area.
(2) Before teaching the robot, check that the robot and its peripheral devices are all in the normal condition.
(3) If it is inevitable to enter the robot operating space to teach the robot, check the locations, settings, and other conditions of the safety devices (such as the EMERGENCY STOP button, the DEADMAN switch on the teach pendant) before entering the area.
(4) The programmer must be extremely careful not to let anyone else enter the robot operating space.
(5) Programming must be done outside of the safety fence as far as possible. If programming needs to be done in the area of the safety fence, the programmer must take the following precautions:
   - Before entering the safety fence area, ensure that there is no risk of hazardous situation in the area.
   - Be ready to press the emergency stop button whenever it is necessary.
   - Operate the Robot at low speed.
   - Before starting programming, check the entire system status to ensure that no remote instruction to the peripheral equipment or motion would harm user .

Our operator panel is provided with an emergency stop button and a key switch (mode switch) for selecting the automatic operation mode (AUTO) and the teach modes (T1 and T2). Before entering the inside of the safety fence for the purpose of teaching, set the switch to a teach mode, remove the key from the mode switch to prevent other people from changing the operation mode carelessly, then open the safety gate. If the safety gate is opened with the automatic operation mode set, the robot stops (Please refer to "**STOP TYPE OF ROBOT**" in SAFETY PRECAUTIONS for detail of stop type). After the switch is set to a teach mode, the safety gate is disabled. The programmer should understand that the safety gate is disabled and is responsible for keeping other people from entering the inside of the safety fence.

Teach pendant is provided with a switch to enable/disable robot operation from teach pendant and DEADMAN switch as well as emergency stop button. These button and switch function as follows:

(1) Emergency stop button: Causes the stop of the robot (Please refer to "STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS for detail of stop type) when pressed.
(2) DEADMAN switch: Functions are different depending on the teach pendant enable/disable switch setting status.
   (a) **Enable:** Servo power is turned off and robot stops when the operator releases the DEADMAN switch or when the operator presses the switch strongly.
   (b) **Disable:** The DEADMAN switch is disabled.
   (Note) The DEADMAN switch is provided to stop the robot when the operator releases the teach pendant or presses the pendant strongly in case of emergency. The R-30*i*B/R-30*i*B Mate employs a 3-position DEADMAN switch, which allows the robot to operate when the 3-position DEADMAN switch is pressed to its intermediate point. When the operator releases the DEADMAN switch or presses the switch strongly, the robot stops immediately.

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

Based on the risk assessment by FANUC, number of operation of DEADMAN switch should not exceed about 10000 times per year.

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

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

T1,T2 mode: DEADMAN switch is effective.

(6) To start the system using the operator’s panel, make certain that nobody is in the robot operating space and that there are no abnormal conditions in the robot operating space.

(7) When a program is completed, be sure to carry out the test operation according to the following procedure.
   (a) Run the program for at least one operation cycle in the single step mode at low speed.
   (b) Run the program for at least one operation cycle in the continuous operation mode at low speed.
   (c) Run the program for one operation cycle in the continuous operation mode at the intermediate speed and check that no abnormalities occur due to a delay in timing.
   (d) Run the program for one operation cycle in the continuous operation mode at the normal operating speed, and check that the system operates automatically without trouble.
   (e) After checking the completeness of the program through the test operation above, execute it in the automatic operation mode.

(8) While operating the system in the automatic operation mode, the teach pendant operator must leave the safety fence.

3.3 SAFETY OF THE MAINTENANCE ENGINEER

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

(1) Must never be in the area during its operation.

(2) A hazardous situation may occur when the robot or the system, are kept with their power-on during maintenance operations. Therefore, for any maintenance operation, the robot and the system must be put into the power-off state. If necessary, a lock should be in place in order to prevent any other person from turning on the robot and/or the system. In case maintenance needs to be executed in the power-on state, the emergency stop button must be pressed.

(3) If it becomes necessary to enter the robot operation area while the power is on, press the emergency stop button on the operator panel, or the teach pendant before entering the area. The maintenance personnel must indicate that maintenance work is in progress and be careful not to allow other people to operate the robot carelessly.

(4) When entering the area enclosed by the safety fence, the maintenance engineer must check the entire system in order to make sure that there is no dangerous situation around. In case the worker needs to enter the safety area whilst a dangerous situation exists, extreme care must be taken, and entire system status must be carefully monitored.

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

(6) Before the start of maintenance, check the robot and its peripheral devices are all in the normal condition.

(7) Do not operate the robot in the automatic mode while anybody is in the robot operating space.

(8) In maintaining the robot parallel to a wall or instrument, or when multiple workers are working nearby, make certain that their escape path is not obstructed.

(9) When a tool is mounted on the robot, or any moving device other than the robot is installed, such as belt conveyor, careful attention required for those motion.

(10) Assign an expert near the operator panel who can press the EMERGENCY STOP button whenever he sees the potential danger.

(11) In case of replacing a part, please contact your local FANUC representative. Wrong procedure may cause the serious damage to the robot and the worker.

(12) Make sure that no impurity into the system in while (in) replacing or reinstalling components.

(13) Turn off the circuit breaker to protect again electric shock in handling each unit or printed circuit board in the controller during inspection. If there are two cabinets, turn off the both circuit breaker.

(14) A part should be replaced with a part recommended by FANUC. If other parts are used, malfunction or damage would occur. Especially, a fuse that is not recommended by FANUC should not be used. Such a fuse may cause not only a damage to the internal parts of the controller but also a fire.

(15) When restarting the robot system after completing maintenance work, make sure in advance that there is no person in the operating space and that the robot and the peripheral devices are not abnormal.

(16) In case of remove the motor or brake, suspend the arm by crane or other equipment beforehand to avoid falling.

(17) Whenever grease is spilled on the floor, remove them as soon as possible to prevent from falling.

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

- Servo motor
- Inside of the controller
- Reducer
- Gearbox
- Wrist unit

(19) Maintenance must be done with appropriate lightning. Be careful that those lightning will not cause any further danger.

(20) When a motor, reducer, or other heavy load is handled, a crane or other equipment should be used to protect maintenance engineers from excessive load. Otherwise, the maintenance engineers would be severely injured.

(21) Must never climb or step on the robot even in the maintenance. If it is attempted, the robot would be adversely affected. In addition, a misstep can cause injury to the worker.

(22) Secure a pedestal and wear the safety belt in performing the maintenance work in high place.

(23) Remove all the spilled oil or water and metal chips around the robot in the safety fence after completing the maintenance.

(24) All the related bolts and components must return to the original place in replacing the parts. If some parts are missing or left (remained), repeat the replacement work until complete the installation.

(25) In case robot motion is required during maintenance, the following precautions should be taken :

- Secure an escape route. And during the maintenance motion itself, monitor continuously the whole system so that your escape route will not become blocked by the robot, or by peripheral equipment.
- Keep vigilant attention for the potential danger. and to press the emergency stop button whenever it is necessary.

(26) Periodic inspection required. (Refer to the robot mechanical manual and controller maintenance manual.) A failure to do the periodical inspection can may adversely affect the performance or service life of the robot and may cause an accident

(27) After replacing some parts, a test run required by the predetermined method. (See TESTING section of “Controller operator’s manual”. During the test run, the maintenance staff must work outside the safety fence.

4 SAFETY OF THE TOOLS AND PERIPHERAL DEVICES

4.1 PRECAUTIONS IN PROGRAMMING

(1) Adopt a limit switch or other sensor to detect a dangerous state and, if necessary, design the program to stop the robot when the sensor signal is received.

(2) Design the program to stop the robot when an abnormal condition occurs in any other robots or peripheral devices, even though the robot itself is normal.

(3) For a system in which the robot and its peripheral devices are in synchronous motion, particular care must be taken in programming in order not to interfere with each other.

(4) Provide a suitable interface between the robot and its peripheral devices so that the robot can detect the states of all devices in the system, and can be stopped according to the states.

4.2 PRECAUTIONS FOR MECHANISM

(1) Keep the component cells of the robot system clean, operate the robot where insulated from the influence of grease, water, and dust.

(2) Don’t use unconfirmed liquid for cutting fluid and cleaning fluid.

(3) Adopt limit switches or mechanical stoppers to limit the robot motion, and avoid the robot from collisions against peripheral devices or tools.

(4) Observe the following precautions about the mechanical unit cables. Failure to follow precautions may cause mechanical troubles.

- Use mechanical unit cable that have required user interface.
- Do not add user cable or hose to inside of mechanical unit.
- Please do not obstruct the movement of the mechanical unit when cables are added to outside of mechanical unit.
- In the case of the model that a cable is exposed, please do not perform remodeling (Adding a protective cover and fix an outside cable more) obstructing the behavior of the outcrop of the cable.
- When installing user peripheral equipment on the robot mechanical unit, please pay attention that equipment does not interfere with the robot itself.

(5) The frequent power-off stop for the robot during operation causes the trouble of the robot. Please avoid the system construction that power-off stop would be operated routinely. (Refer to bad case example.) Please perform power-off stop after reducing the speed of the robot and stopping it by hold stop or cycle stop when it is not urgent. (Please refer to "STOP TYPE OF ROBOT" in SAFETY PRECAUTIONS for detail of stop type.)

(Bad case example)

- Whenever poor product is generated, a line stops by emergency stop and power-off of the robot is incurred.
- When alteration is necessary, safety switch is operated by opening safety fence and power-off stop is incurred for the robot during operation.
- An operator pushes the emergency stop button frequently, and a line stops.
- An area sensor or a mat switch connected to safety signal operates routinely and power-off stop is incurred for the robot.
- Power-off stop is regularly incurred due to an inappropriate setting for Dual Check Safety (DCS).

(6) Power-off stop of Robot is executed when collision detection alarm (SRVO-050) etc. occurs. Please try to avoid unnecessary power-off stops. It may cause the trouble of the robot, too. So remove the causes of the alarm.

5 SAFETY OF THE ROBOT MECHANICAL UNIT

5.1 PRECAUTIONS IN OPERATION

(1) Operating the robot in the jog mode, set it at an appropriate speed so that the operator can manage the robot in any eventuality.
(2) Before pressing the jog key, be sure to comprehend the robot movement by the key in advance.

5.2 PRECAUTIONS IN PROGRAMMING

(1) Design to arrange avoiding mutual interfere when various robot's operation area crossover significantly.
(2) Be sure to specify the predetermined work origin in a motion program so that the robot starts from the origin and terminates at the origin. Make it possible for the operator to distinguish easily that the robot motion has terminated at a glance.

5.3 PRECAUTIONS FOR MECHANISMS

Keep the operating space areas of the robot clean, and operate the robot in an environment free of grease, water, and dust.

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

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

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

6 SAFETY OF THE END EFFECTOR

6.1 PRECAUTIONS IN PROGRAMMING

(1) Circumspect program with sufficient delay required for the program after executing some control command in adopting actuators (pneumatic, hydraulic, and electric)
(2) Adopt limit switches for the end effector, and control the robot system by monitoring the state.

7 STOP TYPE OF ROBOT

There are following three types of Stopping Robot.

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

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

"**Power-Off stop**" performs following processing.

- An alarm is generated, and then the servo power turns off. Instantly the robot stops.
- Execution of the program is paused.

Frequent Power-Off stop of the robot during operation can cause mechanical problems of the robot.

Avoid system designs that require routine or frequent Power-Off stop conditions.

Controlled stop (Category 1 following IEC 60204-1)

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

"**Controlled stop**" performs following processing.

- The alarm "**SRVO-199 Controlled stop**" occurs along with a decelerated stop. The program execution is paused.
- An alarm is generated, and then the servo power turns off.

Hold (Category 2 following IEC 60204-1)

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

"**Hold**" performs following processing.

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

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

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

There are the following 3 Stop patterns.

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

P-Stop:	Power-Off stop
C-Stop:	Controlled stop
-:	Disable

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

Option	R-30*i*B/ R-30*i*B Mate
Standard	A (*)
Controlled stop by E-Stop (A05B-2600-J570)	C (*)

(*) R-30*i*B / R-30*i*B Mate does not have servo disconnect. R-30*i*B Mate does not have SVOFF input.

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

"Controlled stop by E-Stop" option

When "**Controlled stop by E-Stop**" (A05B-2600-J570) option is specified, the stop type of the following alarms becomes Controlled stop but only in AUTO mode. In T1 or T2 mode, the stop type is Power-Off stop which is the normal operation of the system.

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

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

- In Controlled stop, the robot is stopped on the program path. This function is effective for a system where the robot can interfere with other devices if it deviates from the program path.
- In Controlled stop, physical impact is less than Power-Off stop. This function is effective for systems where the physical impact to the mechanical unit or EOAT (End Of Arm Tool) should be minimized.
- The stopping distance and time of Controlled stop is longer than the those of Power-Off stop, depending on the robot model and axis. Please refer to the operator's manual of a particular robot model for the data of stopping distance and time.

When this option is loaded, this function cannot be disabled.

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

⚠ **WARNING**

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

150916

TABLE OF CONTENTS

1 PREFACE

This chapter explains the manual plan.

1.1 ABOUT MANUAL

About this manual

FANUC Robot series (R-30*i*B/R-30*i*B Mate CONTROLLER) Optional Function OPERATOR’S MANUAL.
This manual describes how to operate the FANUC Robot, an all-purpose compact robot. It is controlled by the FANUC R-30*i*B and R-30*i*B Mate controller (called the robot controller hereinafter) containing the FANUC Robot software.

This manual describes the software optional functions. Each chapter describes one software option. Please select and refer to the chapters describing your required function.

Related manuals

The following manuals are available:

Robot controller	Optional Function OPERATOR’S MANUAL **B-83284EN-2** (This manual)	Topics: Description of the software optional functions. Use: Guide to teaching, introducing, and adjusting the robot at the work site, and application designing.
	OPERATOR’S MANUAL (Basic Operation) **B-83284EN**	Topics: Functions, operations and the procedure for operating the robot. Programming procedure, interface and alarm. Use: Guide to teaching, introducing, and adjusting the robot at the work site, and application designing.
	OPERATOR'S MANUAL (Alarm Code List) **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.
	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.
	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),** **B-83525EN (for R-30*i*B Mate)** **B-83555EN (for R-30*i*B Mate Open Air)**	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 manual describes the following software options.

Chapter 3	SOFT FLOAT FUNCTION
Chapter 4	CONTINUOUS ROTATION FUNCTION
Chapter 5	OPERATION GROUP DO OUTPUT FUNCTION
Chapter 6	AUTOMATIC ERROR RECOVERY FUNCTION
Chapter 7	REMOTE TCP FUNCTION
Chapter 8	HIGH SENSITIVITY COLLISION DETECTION
Chapter 9	LOAD ESTIMATION
Chapter 10	PAYLOAD OVER AND PAYLOAD SETTING CONFIRM FUNCTION
Chapter 11	GRAVITY COMPENSATION
Chapter 12	OPERATION LOG BOOK
Chapter 13	PROGRAM TOOLBOX
Chapter 14	ADVANCED CONSTANT PATH
Chapter 15	AUTO SINGULARITY AVOIDANCE FUNCTION
Chapter 16	PATH SWITCHING FUNCTION
Chapter 17	AUXILIATY AXIS SERVO OFF (LOCAL STOP) FUNCTION
Chapter 18	DUAL MOTOR DRIVE
Chapter 19	MULTI UOP INTERFACE FUNCTION
Chapter 20	ERROR CODE OUTPUT FUNCTION
Chapter 21	DATA MONITOR FUNCTION
Chapter 22	BRAKE CHECK FUNCTION
Chapter 23	PANEL WIZARD
Chapter 24	ENHANCED MIRROR IMAGE
Chapter 25	CUSTOMIZE SUPPORT FUNCTION
Chapter 26	KAREL PROGRAM EXECUTION HISTORY RECORD
Chapter 27	TORQUE LIMIT FUNCTION
Chapter 28	TCP SPEED OUTPUT INSTRUCTION
Chapter 29	TP DRAM/FILE STORAGE FUNCTION
Chapter 30	CYCLE TIME TRACKING
Chapter 31	MATH FUNCTION INSTRUCTION
Chapter 32	SERVO TOOL CHANGE FUNCTION
Chapter 33	OPERATION WITHOUT SHIFT FUNCTION
Chapter 34	HIGH SPEED SHIFT FUNCTION
Chapter 35	*i*RDiagnostics
Chapter 36	MENU UTILITY FUNCTION
Chapter 37	4D GRAPHICS FUNCTION
Chapter 38	DATA TRANSFER BETWEEN ROBOTS FUNCTION
Chapter 39	TOUCH SENSING
Chapter 40	TOUCH SKIP FUNCTION
Chapter 41	MROT INSTRUCTION
Chapter 42	ROBOT ISOLATION FUNCTION
Chapter 43	ANTI-DEFLECTION FOR EXTERNAL FORCE
Chapter 44	INTERFACE PANEL FUNCTION
Chapter 45	SPECIAL JOG SEQUENCE
Chapter 46	MOTION INSTRUCTION ENHANCED EDITING
Chapter 47	FINISHING FUNCTION PACKAGE
Chapter 48	JOINT POSITION OUTPUT FUNCTION
Chapter 49	EXPANDED REGISTERS FUNCTION
Chapter 50	STITCH FUNCTION
Chapter 51	VISUAL DIAGNOSTICS

Chapter 52	PDF VIEWER FUNCTION
Chapter 53	HELP AND DIAGNOSTICS DISPLAY
Chapter 54	MAINTENANCE REMINDER
Chapter 55	HMI DEVICE COMMUNICATION
Chapter 56	FAULT & INCIDENT REPORT

3 SOFTFLOAT FUNCTION

Usually, the robot moves accurately toward the goal specified using the teach pendant (taught point).
When the robot is used to mount workpieces on a machine tool, variances in workpiece precision may result in a shift in the workpiece position relative to the tool, thus possibly causing interference between the workpiece and tool.
A softfloat function has been added which is effective in mounting workpieces with variances in precision onto a machine tool.
The softfloat function is also very effective if the synchronization speed is unstable as in the extraction of workpieces in sync with hydraulic extrusion, and if workpieces that the robot cannot grip accurately, such as rough-machined workpieces, are to be handled.

3.1 JOINT / CARTESIAN SOFTFLOAT

Function

The joint / Cartesian softfloat function works as follows:

- Two types of softfloat are supported: joint softfloat for specifying the softness related to the direction of rotation of each arm of the robot, and Cartesian softfloat for specifying the softness on the Cartesian axes.
- The function is enabled/disabled using an instruction in the program. Its conditions are also specified using the instruction.
- ”Servo flexibility” can be specified for each axis. The term servo flexibility indicates how strongly the axis resists external forces. It is specified between 0% and 100%. A servo flexibility of 100% corresponds to being the most flexible. The servo flexibility is specified using a condition table that contains a set of data for one group (for nine axes).
- If an external force above a certain level (so high as to overcome a static frictional force) is applied to a robot, the axis of the robot is pressed and moved.
- An external force applied to a robot may prevent it from reaching the taught point. The distance between the taught point and the point the robot can reach is nearly proportional to the magnitude of the external force.
- If static load is applied to a robot, the robot controls force to maintain its attitude even if the softfloat function is enabled.

The detailed descriptions of the softfloat function follow.

Program instruction

The following three program instructions related to the softfloat function are supported.

- SOFTFLOAT[n]

The softfloat function is enabled using condition n.
* The setting of softfloat condition is explained in ”Condition setting menu”.

- SOFTFLOAT END

The softfloat function is disabled.

- FOLLOW UP

When an external force is removed from a robot, it usually tries to go back to the taught point. However, this instruction causes the robot to assume that the current position is the taught point, and prevents it from going back to the taught point.

Softfloat function effective range

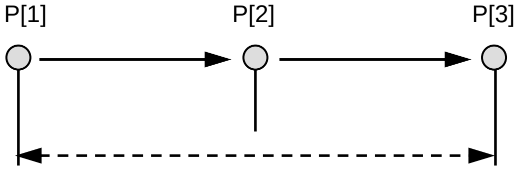
The SOFTFLOAT[n] instruction can be used in two modes; in one mode it is used solely in a program and in the other mode it is used as an auxiliary motion instruction after a motion statement. The range in which the softfloat function is effective for robot operation is determined according in to which mode this instruction is used.

- Sole instruction

The softfloat function is enabled after the end of the motion specified on the line preceding the solely specified SOFTFLOAT[n] instruction.

In the following example, the softfloat function is enabled after the motion specified on line 1 ends, and disabled by SOFTFLOAT END on line 5.

```
1: J P[1] 100% FINE
2:  SOFTFLOAT[1]
3: L P[2] 100mm/sec FINE
4: L P[3] 100mm/sec FINE
5:  SOFTFLOAT END
```



The soft float function is enabled.

- Auxiliary motion instruction

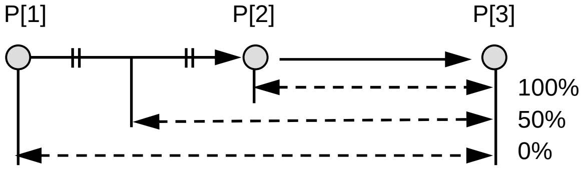
The softfloat function becomes enabled during execution of a motion statement attached with a SOFTFLOAT [n] instruction.

The point at which the softfloat function becomes enabled is determined by a softfloat condition ”Exec Start Ratio”.

Auxiliary motion instruction is specified as the ratio (from 0% to 100% in 1% steps) of a distance to be traveled before the robot reaches the taught point corresponding to a motion statement attached with a SOFTFLOAT[n].

In the following example, the softfloat function is effective between P[1] taught using a motion statement on line 1 and P[2] taught using a motion statement on line 2 attached with the SOFTFLOAT[n] instruction.

```
1: J P[1] 100% FINE
2: L P[2] 100mm/sec FINE SOFTFLOAT[1]
3: L P[3] 100mm/sec FINE
4: SOFTFLOAT END
```



The soft float function is enabled.

NOTE

The auxiliary motion instruction is not supported by Cartesian softfloat.

- Condition setting menu

The softfloat conditions are specified on the [SETUP Softfloat] menu, which consists of the following two menus.

- List menu

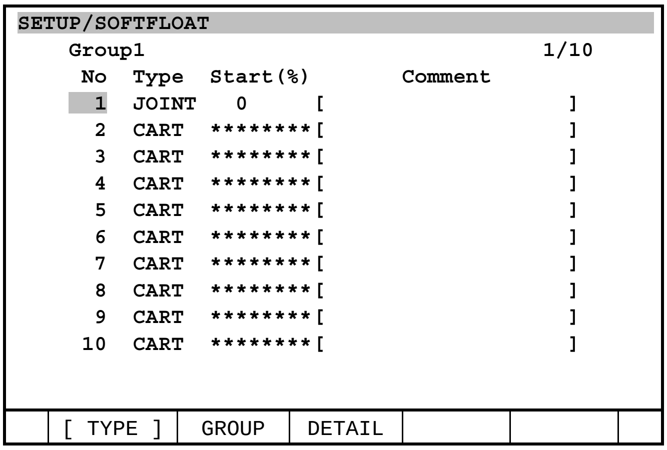
- Detail menu

A function key is used to select either menu.

- Pressing the F3, DETAIL key on the list menu selects the detail menu.
- Pressing the F3, LIST key on the detail menu selects the list menu.

Up to 10 conditions can usually be specified for the softfloat function.

List menu

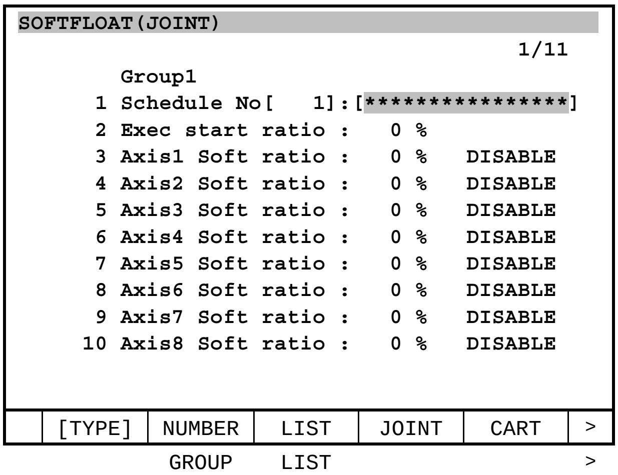


```
SETUP/SOFTFLOAT
    Group1                                1/10
     No  Type  Start(%)       Comment
      1  JOINT   0      [                   ]
      2  CART  ********[                   ]
      3  CART  ********[                   ]
      4  CART  ********[                   ]
      5  CART  ********[                   ]
      6  CART  ********[                   ]
      7  CART  ********[                   ]
      8  CART  ********[                   ]
      9  CART  ********[                   ]
     10  CART  ********[                   ]

   [ TYPE ]  GROUP   DETAIL
```

F3, DETAIL

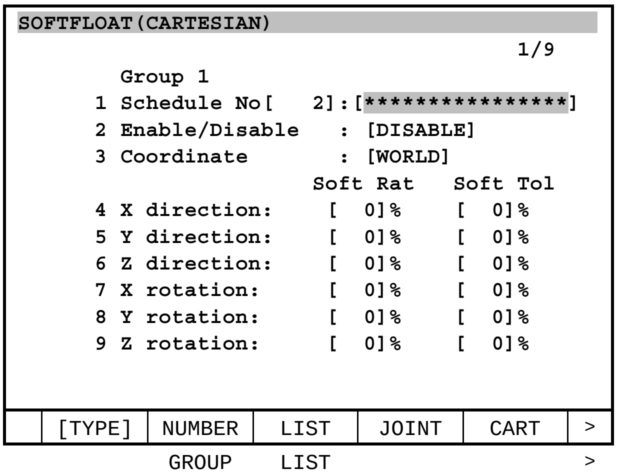
F3, LIST

Joint softfloat screen



```
SOFTFLOAT(JOINT)
                                           1/11
        Group1
      1 Schedule No[   1]:[****************]
      2 Exec start ratio :   0 %
      3 Axis1 Soft ratio :   0 %   DISABLE
      4 Axis2 Soft ratio :   0 %   DISABLE
      5 Axis3 Soft ratio :   0 %   DISABLE
      6 Axis4 Soft ratio :   0 %   DISABLE
      7 Axis5 Soft ratio :   0 %   DISABLE
      8 Axis6 Soft ratio :   0 %   DISABLE
      9 Axis7 Soft ratio :   0 %   DISABLE
     10 Axis8 Soft ratio :   0 %   DISABLE

  [TYPE]  NUMBER  LIST   JOINT   CART   >
          GROUP   LIST                  >
```

Cartesian softfloat screen

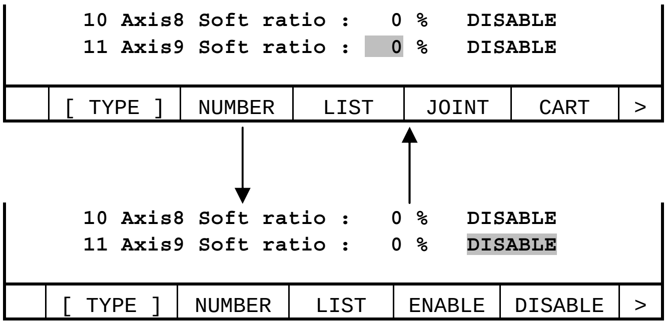
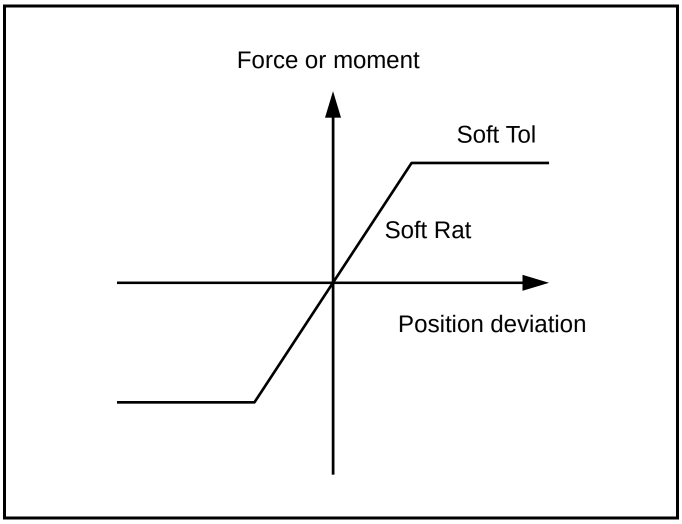


```
SOFTFLOAT(CARTESIAN)
                                            1/9
        Group 1
      1 Schedule No[   2]:[****************]
      2 Enable/Disable    : [DISABLE]
      3 Coordinate        : [WORLD]
                       Soft Rat   Soft Tol
      4 X direction:     [  0]%    [  0]%
      5 Y direction:     [  0]%    [  0]%
      6 Z direction:     [  0]%    [  0]%
      7 X rotation:      [  0]%    [  0]%
      8 Y rotation:      [  0]%    [  0]%
      9 Z rotation:      [  0]%    [  0]%

  [TYPE]  NUMBER  LIST   JOINT   CART   >
          GROUP   LIST                  >
```

The following data can be specified on the detail menu.

Table 3.1 Setting items of softfloat detail menu

Items	Descriptions
Condition No. / Comment	Softfloat condition number. By default, ten numbers can be set. Pressing the [ENTER] key with the cursor on line 1 enables entering a comment. The comment text can be specified in the same way as on other menus.
Exec start ratio	Line 2 specifies the point where the softfloat function is enabled if the SOFTFLOAT [n] is used as an auxiliary motion instruction. See ”Softfloat function effective range” for the softfloat start ratio.

Items	Descriptions
Soft ratio	Soft ratio for each axis can be specified on line 3 and the subsequent lines. The soft ratio indicates how strongly the axis resists external forces. It is specified between 0% and 100%. A soft ratio of 100% corresponds to being the most flexible. Whether the softfloat function is enabled/disabled can be specified for each axis on line 3 and the subsequent lines. Setting the cursor at the rightmost end (enabled/disabled setting position) of each line causes the F4 (ENABLE) and F5 (DISABLE) keys to appear. Use these keys to specify whether to enable/disable the softfloat function. **NOTE** Pressing the F2 (NUMBER) key selects another page of the detail menu for other conditions.  10 Axis8 Soft ratio : 0 % DISABLE 11 Axis9 Soft ratio : 0 % DISABLE [TYPE] NUMBER LIST JOINT CART > 10 Axis8 Soft ratio : 0 % DISABLE 11 Axis9 Soft ratio : 0 % DISABLE [TYPE] NUMBER LIST ENABLE DISABLE >
Enable/Disable	When this item is set to DISABLE, softfloat cannot be executed.
Coordinate	Select one of WORLD, USER, and TOOL. **NOTE** If the remote TCP is used, USER indicates the coordinate system on the remote TCP.
X direction etc.	Set the softness on or around the X-, Y-, and Z-axes. If Soft Rat increases, the spring constant decreases, allowing the robot to move with less force. If Soft Tol increases, the maximum force and moment applied by the robot in that direction decreases, allowing the robot to move with less force. The difference between Soft Rat and Soft Tol is illustrated below.  Force or moment Soft Tol Soft Rat Position deviation

Operation area limitation

Operation area during Cartesian softfloat is able to be limited.

If the position error between TCP of motion command and actual TCP in the frame which is set in softfloat schedule exceeds the threshold, alarm occurs and robot stops.

For safety, following system variables have to be set to avoid interference before the program which executes Cartesian softfloat runs.

To reflect new value of these system variables, power OFF/ON is necessary.

- $param_group[g].$cb_mass =1.0 :enable limitation
- $param_group[g].$cb_ix =Threshold in X direction [mm]
  (0.0 occurs alarm)
- $param_group[g].$cb_iy =Threshold in Y direction [mm]
  (0.0 occurs alarm)
- $param_group[g].$cb_iz =Threshold in Z direction [mm]
  (0.0 occurs alarm)

3.2 PUSHOUT SOFTFLOAT

Pushout softfloat is suitable for the extraction of workspaces in sync with hydraulic extrusion.
Each softfloat type has the following features.
Each softfloat schedule can select different softfloat type.

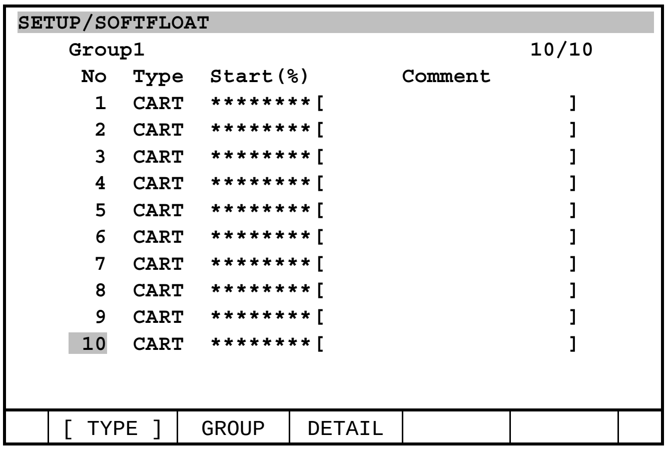
- Joint softfloat
  Softness around robot joint can be specified.
- Cartesian softfloat
  Softness along Cartesian coordinate and around Cartesian coordinate can be specified.
  It is good at contouring and face matching with robot motion.
- Pushout softfloat
  In this softfloat, force and moment necessary become smaller than current Cartesian softfloat.
  Softness around Cartesian coordinate can not be specified.
  This softfloat does not work in where J5 = -3.5~+3.5 [deg].
  It is good at following the pushout.

Procedure 3-1 Procedure to show pushout softfloat screen

Select "PUSH" in softfloat detail screen.

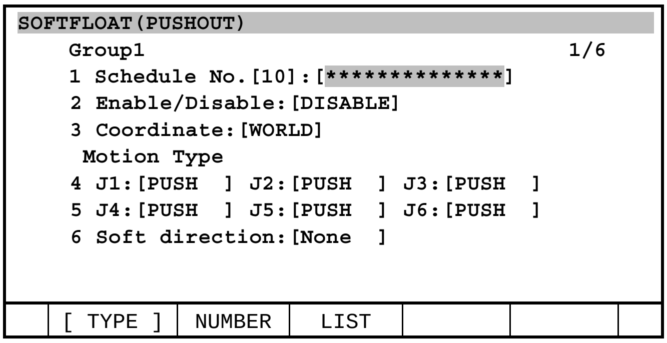
Step

1 Press [MENU] key to show screen list.
2 Select "6 SETUP".
3 Push F1, [TYPE] key to show screen list.
4 Select "softfloat". Softfloat list screen will be displayed.



```
SETUP/SOFTFLOAT
    Group1                                  10/10
     No  Type   Start(%)        Comment
      1  CART   ********[                    ]
      2  CART   ********[                    ]
      3  CART   ********[                    ]
      4  CART   ********[                    ]
      5  CART   ********[                    ]
      6  CART   ********[                    ]
      7  CART   ********[                    ]
      8  CART   ********[                    ]
      9  CART   ********[                    ]
     10  CART   ********[                    ]

    [ TYPE ]   GROUP    DETAIL
```

5 Set cursor on schedule No. to set pushout softfloat, and push F3, DETAIL key.
6 Push NEXT key until "PUSH" are shown on function key line.
7 Push F4, PUSH key. Detail screen of pushout softfloat will be displayed.



```
SOFTFLOAT(PUSHOUT)
    Group1                                      1/6
    1 Schedule No.[10]:[**************]
    2 Enable/Disable:[DISABLE]
    3 Coordinate:[WORLD]
     Motion Type
    4 J1:[PUSH  ] J2:[PUSH  ] J3:[PUSH  ]
    5 J4:[PUSH  ] J5:[PUSH  ] J6:[PUSH  ]
    6 Soft direction:[None  ]

   [ TYPE ]  NUMBER   LIST
```

Procedure 3-2 Procedure to set pushout softfloat

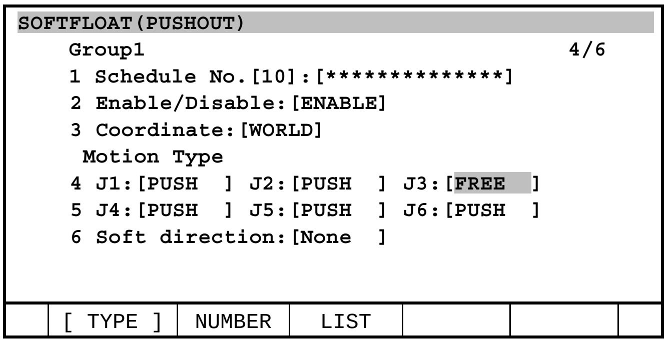
In addition to the setup of Cartesian softfloat, “motion type” and “soft direction” should be set. “Soft rat” and ”soft tol” are removed.

Step

1 Set “enable/disable” and “coordinate” same as current softfloat.

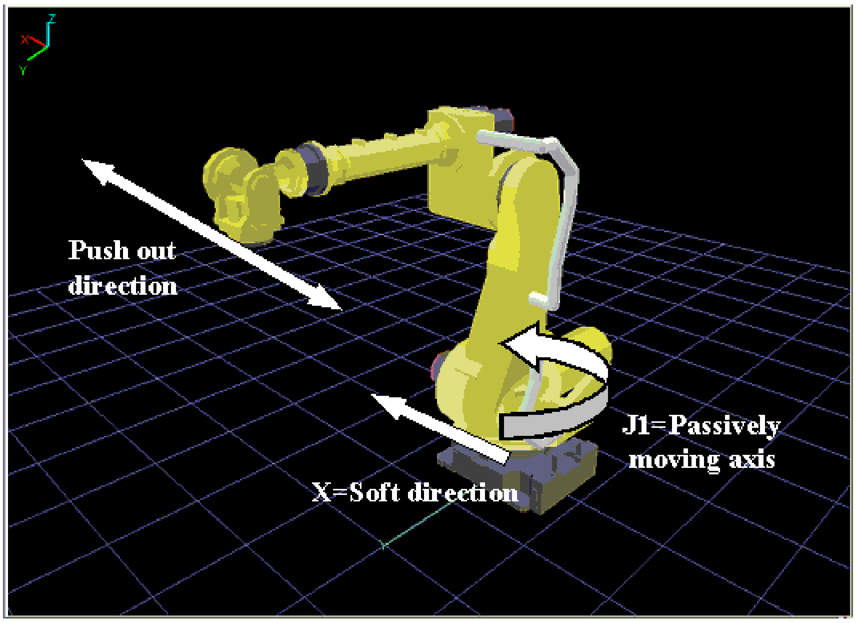
2 Set “soft direction” to the direction which should be soft in the schedule No.

3 Set motion type of the axis which should move passively following the force from outside to “FREE”. Only one axis can be “FREE” in one group.



```
SOFTFLOAT(PUSHOUT)
    Group1                                      4/6
    1 Schedule No.[10]:[**************]
    2 Enable/Disable:[ENABLE]
    3 Coordinate:[WORLD]
     Motion Type
    4 J1:[PUSH  ] J2:[PUSH  ] J3:[FREE  ]
    5 J4:[PUSH  ] J5:[PUSH  ] J6:[PUSH  ]
    6 Soft direction:[None  ]

   [ TYPE ]  NUMBER   LIST
```

Example)
When J1 is 90 degree, to make X direction of world coordinate soft, motion type of J1 should be “FREE”.



Procedure 3-3 Procedure to use pushout softfloat

Procedure is the same as that of joint/Cartesian softfloat function.

Step

1 In edit screen, open the program to use pushout softfloat.
2 Push NEXT key until [INST] is shown on function key.
3 Teach softfloat[…] instruction on the line to start pushout softfloat.
4 Set softfloat schedule No. of pushout softfloat in the index of softfloat[…] instruction.
5 Teach softfloat end instruction on the line to stop pushout softfloat.

```
1: J P[1] 100% FINE
2: SOFTFLOAT[10]
3: WAIT 10.00(sec)
4: SOFTFLOAT END
```

3.3 KNOWHOW OF SOFTFLOAT

Timing to start softfloat

Force from outside should not be added when softfloat starts.
If softfloat starts with not only gravity force but also force from outside, robot will move above or below, and "Stop/Move error excess" or "Softfloat time out" alarm will occur.
For example in pushout handing application, you should make program as that softfloat starts before hand closed.
This know-how is effective in all softfloat of joint, Cartesian and pushout.

Bad example)

```
1: CALL HNDCLOSE
2: SOFTFLOAT[10]
3: WAIT 10.00(sec)
4: SOFTFLOAT END
```

Force from outside will be added if softfloat starts after hand closed.

Good example)

```
1: SOFTFLOAT[10]
2: CALL HNDCLOSE
3: WAIT 10.00(sec)
4: SOFTFLOAT END
```

Force from outside will not be added if softfloat starts before hand closed.

Know-how to improve softness

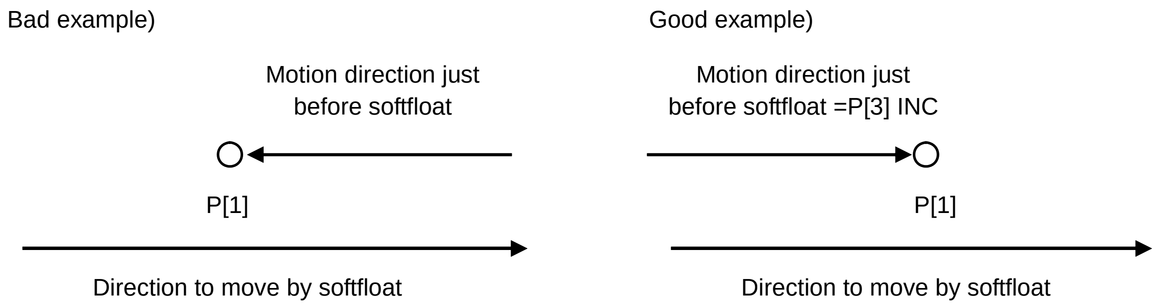
Even with the same softfloat schedule setting, the softness of softfloat differs depending on the direction of the motion just before softfloat starts.
It is caused by the feature of that softfloat is more soft to move in the same direction to the motion just before and less soft to move in the opposite direction to the motion just before.
So, you should make program as that just before softfloat starts, move robot a little in the direction opposite to the direction robot should move by softfloat, then move robot back to the position to start softfloat. The distance necessary is 1~2[mm] or longer.
For example, when you want to start softfloat to move in +X direction in P[1], you should add the following motion instruction between the motion instruction to P[1] and softfloat instruction.

```
1: J P[1:sflt start pos] 100% FINE
2: L P[2:-x1mm] 100mm/sec CNT0 INC
3: L P[3:+x1mm] 100mm/sec CNT0 INC
4: SOFTFLOAT[10]
5: CALL HNDCLOSE
6: WAIT 10.00(sec)
7: SOFTFLOAT END
```

To go back to P[1] in the next line, move opposite direction in this line.

Move back to P[1] in the the direction to move by softfloat.

By this method, robot will become more soft to move +X direction and less soft to move –X direction. This know-how is effective in all softfloat of joint, Cartesian and pushout.

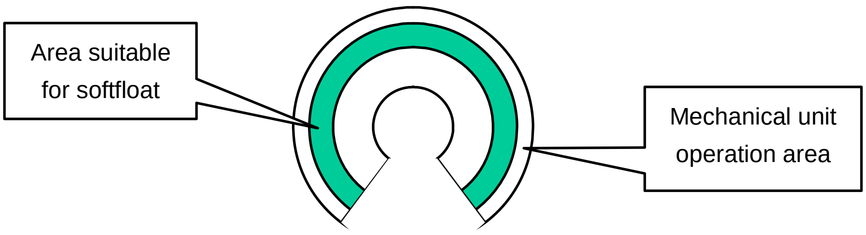

Bad example)
Motion direction just before softfloat
P[1]
Direction to move by softfloat
Good example)
Motion direction just before softfloat =P[3] INC
P[1]
Direction to move by softfloat

3.4 KNOWHOW OF CARTESIAN SOFTFLOAT

Softness along Cartesian coordinate and around Cartesian coordinate can be specified.
It is good at contouring and face matching with robot motion.
Know-how to use it effectively will be explained.

All application common know-how

- Layout and tool selection
  Area in 10~20[%] inside from the outside border of mechanical unit operation area is recommended. And select the tool as short as possible.



- Schedule setup
  Set 100% in “soft rat” of the direction or rotation that you want to make soft.
  Adjust “soft tol” of pushing direction to limit force to avoid pushing with too large force.

- Velocity teaching
  In softfloat with robot motion, robot may keep still with small velocity taught.
  Teach the velocity more than about 100 mm/s.

Contouring application know-how

- Schedule setup
  Example) In case pushing Z direction

	Soft Rat		Soft Tol		
X direction	100	%	0	%	Make X, Y direction soft.
Y direction	100	%	0	%	
Z direction	0	%	70	%	Limit force to avoid pushing with too large force.
X rotation	0	%	0	%	Make rotation hard.
Y rotation	0	%	0	%	
Z rotation	0	%	0	%	

- TCP setup
  In application which should keep tool direction, TCP should be X, Y and Z =0 when softfloat starts. By this, difference between soft direction and hard direction appears well.

Example)

```
1: UTOOL_NUM=9
2: SOFTFLOAT[10]
3: UTOOL_NUM=2
4: WAIT 10.00(sec)
5: SOFTFLOAT END
```

Tool frame whose X=0, Y=0, Z=0 and W, P, R are the same as actual tool

TCP set as actual tool

Face matching application know-how

- Schedule setup
  Example) In case pushing Z direction

	Soft Rat		Soft Tol		
X direction	0	%	0	%	Make X, Y direction hard.
Y direction	0	%	0	%	
Z direction	0	%	70	%	Limit force to avoid pushing with too large force.
X rotation	100	%	0	%	Make rotation soft.
Y rotation	100	%	0	%	
Z rotation	100	%	0	%	

- TCP setup
  Use the tool frame set as actual tool. You don’t have to change TCP when softfloat starts.
  Select the tool as short as possible. X, Y and Z direction can’t become hard enough if tool is long.
  Moment of the direction to match face is necessary. Select tool that can generate moment of the direction to match face by pushing even with largest orientation mismatch.

3.5 CAUTIONS / RESTRICTIONS

When using the softfloat function, observe the following cautions / restrictions.

- Restrictions imposed when the softfloat function is enabled
  - It is not guaranteed that the robot always follows the taught path.
  - The taught route changes according to override.
  - The required operation time may be prolonged compared with normal operation.
- The softfloat function is disabled automatically when:
  - Program execution starts.
  - Program execution ends.
  - The program stops due to an alarm that turns off the servo.
  - Jog feed is performed with the program at pause.
  - The program is restarted after the cursor is moved manually with the program at pause.
  - Backward execution is performed.
  - Power is applied.

- If the program is caused to pause, then restarted, the states of the softfloat function (such as enabled/disabled and the softfloat start ratio) are set to the conditions which exist before the program is caused to pause. However, the softfloat function is disabled if the operation listed above is done.
- The softfloat function cannot be enabled by any method other than the SOFTFLOAT instruction.
- When the softfloat function is enabled, the robot moves in the CNT 0 mode (no position check is made) even if FINE has been specified as motion statement positioning mode.
- When the softfloat function is enabled, if an external force causes the robot to move beyond a certain distance, the following servo alarms occur.
  - If the robot is at rest :

  [SRVO-023 Stop error excess(G:i A:j)]
  - If the robot is operating :

  [SRVO-024 Move error excess(G:i A:j)]
- If an attempt is made to enable the softfloat function with a brake applied, the brake is released automatically before the function is enabled.
- When the softfloat function is enabled, brake control is ineffective.
- If the motion group mask in a program is [*,*,*,*,*,*,*,*] (there is no motion group), when the program issues instructions with the softfloat function, the following alarm occurs:
  [INTP-216 (program name, line number) Invalid value for group number]
- The range of motion with the softfloat function enabled should be minimized. A weight balance may vary depending on the softfloat ratio and travel distance, thus shifting the vertical axis upward or downward.
  The range of motion with an auxiliary motion instruction issued should also be minimized for the same reason. In addition, the speed of motion should be kept low.
- When the softfloat function is enabled, follow-up processing is normally performed for individual motion instructions.
  This processing is enabled or disabled according to system variable $SFLT_DISFUP.

  $SFLT_DISFUP Default value: FALSE
  - If FALSE, follow-up is performed at the start of each motion instruction in the program.
  - If TRUE, follow-up is not performed for individual motion instructions in the program.
- This function cannot be used with arc tools.
- Force from outside should not be added when softfloat starts, when softfloat ends and when follow up is doing.
  If softfloat starts with not only gravity force but also other force from outside, robot will move above or below, and “Stop/Move error excess” or “Softfloat time out” alarm will occur.
  In such cases, please modify TP program by changing position data or by adding wait instruction as softfloat starts and ends without other force.

NOTE

Follow-up

With the softfloat function, external forces are applied to the robot so that it operates at positions slightly different from those specified. When the external force is removed after the completion of the operation, the robot usually attempts to move back to a specified point abruptly. Follow-up prevents this abrupt movement.

4 CONTINUOUS ROTATION FUNCTION

The continuous rotation function allows continuous and limitless rotation about the final axis or an additional rotation axis of the robot in one direction.

NOTE
For example, the ”final axis” refers to the J6 axis of a robot having six axes.

For example, this function is useful for rotating those devices that require continuous rotation, such as conveyers, pumps, and grinders, about a robot axis or additional rotation axis.

To specify the items for this function, such as disable/enable, use the SETUP Continuous Turn screen. The start and stop of continuous rotation are directed from a program.

Before this function can be used, the setup necessary for continuous rotation must be performed.

Only a single continuous rotation axis can be allocated for each operation group. The axis must satisfy the following conditions:

- Final axis of the robot
- Final axis of the built-in additional rotation axes
- Any of the normal additional rotation axes
- Final axis of the independent additional axes

The continuous rotation axis must satisfy the following mechanical conditions:

- The mechanism must allow continuous operation (must be free of obstacles such as stoppers).
- The gear reduction ratio (value of Gear Ratio (motor) / Gear Ratio (axis) on the setting screen, the speed of the motor required for one rotation about the axis) must be 4000 or less.

To use this function, an option (continuous rotation function) is required.

Function

When this function is enabled, the axis allocated as a continuous rotation axis allows limitless rotation. The angle on the axis is, therefore, represented by a relative degree within +180°, not by an absolute one. For example, the figure below shows rotation from 0° to 200° in the positive direction. The angle on the axis after the rotation is -160°, not 200°.

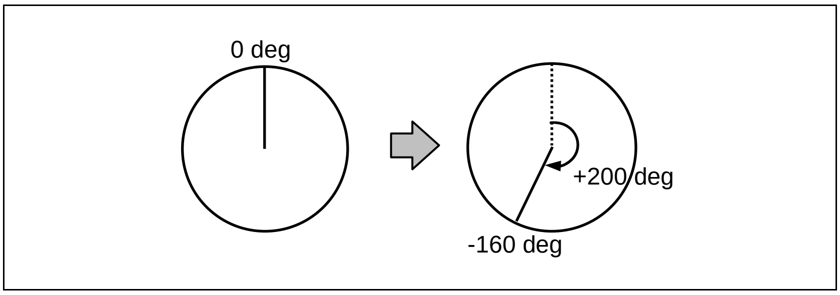


Fig. 4 (a) Angle on the Continuous Rotation Axis

When this function is enabled but continuous rotation is not performed (see the next page for an explanation of how to use continuous rotation), rotation is performed about the continuous rotation axis from the current angle to the target angle in whichever direction incurs the least amount of motion. (Usually, the direction of rotation about the axis is determined with the relationship between the current and target angles.) This ”shorter-way operation” is effective in reducing the cycle time.

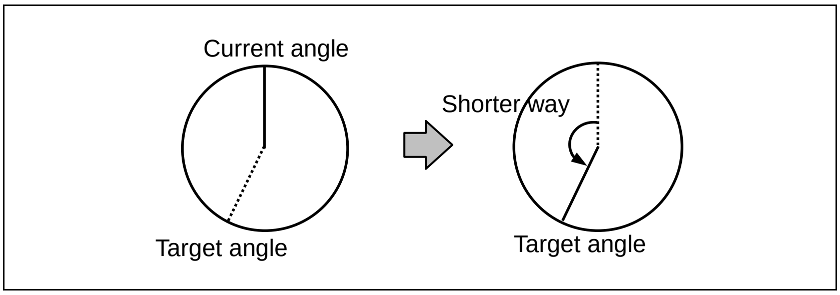


Fig. 4 (b) Shorter-Way Operation

Setup

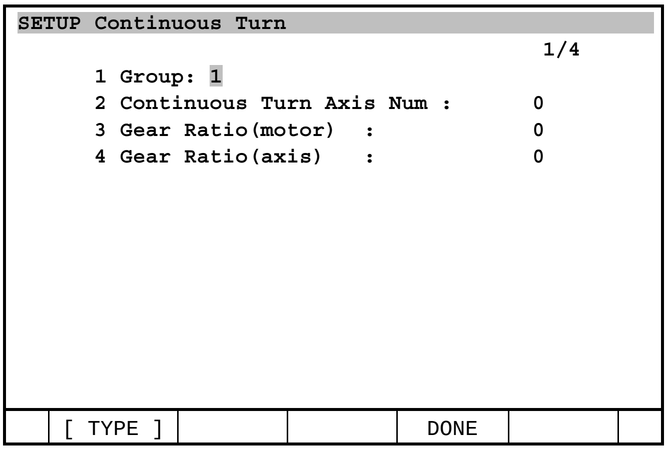
To use the function,

- Perform setup on the SETUP Continuous Turn screen, and
- Specify the start/stop of continuous rotation with the operation add instruction, ”continuous rotation speed instruction”.

Procedure 4-1 Setting up the continuous rotation function

Step

1 Press [MENU] key. The screen menu will be displayed.
2 Select SETUP.
3 Press F1, [TYPE]. The screen switching menu will be displayed.
4 Select Cont Turn. The continuous rotation setup screen will be displayed.



```
SETUP Continuous Turn
                                              1/4
      1 Group: 1
      2 Continuous Turn Axis Num :       0
      3 Gear Ratio(motor)  :             0
      4 Gear Ratio(axis)   :             0
```

5 Specify the necessary items using the numeric and other keys.
- To disable the continuous rotation function, set ”0” for Continuous Turn Axis Num.
- The maximum value for Gear Ratio (motor) and Gear Ratio (axis) is 2147483646.
- Set the operation group number for Group. If a different number (number of the operation group to be viewed) is entered in this field, the other settings are changed to those of the operation group.

6 After specifying the items, press F4, DONE.
7 Set up Constant Path function. For detail about simultaneous use of Continuous Rotation and Constant Path function, please refer to next section “Use of Continuous Rotation function with Constant Path function”.
- In case that Constant Path function is needed to be disabled by limitation (Example: Continuous Rotation axis is set in final axis of robot), Constant Path function is disabled and message “ConstPath is disabled.(Press ENTER)” is displayed.
- In case that the user can choose set Constant Path disabled or not (Example: Continuous Rotation axis is set in independent axis), “Disable Constant Path Function?” is displayed.
  - When F2, YES is pressed, Constant Path function is disabled and “ConstPath is disabled.(Press ENTER)” is displayed.

- When F3, NO is pressed, setup of Constant Path function is NOT changed. If Constant Path function is enabled, “ConstPath is enabled.(Press [ENTER] key)” is displayed. Else if Constant Path function is disabled, “ConstPath is disabled.(Press [ENTER] key)” is displayed.

8 If [ENTER] key is pressed, message “Cycle power to take effect (press [ENTER] key)” will be displayed.

9 Turn off the power, then turn it back on with a cold start. (At first cycle power from Continuous Rotation axis setup, cycle power will be done by cold start forcedly. Later second cycle power, cycle power will be done by selected start mode.)

The items on the continuous rotation setup screen are described below.

Table 4 Contents of the Continuous Rotation Setup Screen

Item	Description
Group	Set the operation group number.
Continuous Turn Axis Num	Set the number of the continuous rotation axis. If ”0” is set, this function is disabled for the operation group.
Gear Ratio (motor) Gear Ratio (axis)	Set the gear reduction ratio for the continuous rotation axis set for the above item. A value from 0 to 2147483646 can be set for each item. The items must, however, satisfy the following: Gear Ratio (motor) ÷ Gear Ratio (axis) ≤ 4000

Use of Continuous Rotation function with Constant Path function

- Continuous rotation axis group cannot support Constant Path function.
- In case that Continuous Rotation axis is set in independent axis and group mask is separated from non-Continuous Rotation group, Constant Path can support only non-Continuous Rotation groups.
- For example, following program can support simultaneous use of Continuous Rotation function with Constant Path function.

[Example]

- G1 : Robot, G2 : Servogun, G3 : Continuous Rotation axis (Independent axis with 1 axis)
  - Program 1 : G1 + G2 (Robot + Servogun) --- < 1, 1, * >
  - Program 2 : G3 (Continuous Rotation axis) only --- < *, *, 1 >
- In Program1, Constant Path and Cycle Time Priority are active.
- In Program2, Continuous Rotation can work in a program.
- To combine Constant Path and Cycle Time Priority with continuous rotation, run Program2 in Program1 concurrently.

- In multi-group program, if continuous rotation axis exists in any of the groups, alarm “CPMO-004 Feature not Supported (G:%d^2) ” will be posted at execution. To use continuous rotation in a program, please separate groups which have Continuous Rotation axis from groups that DOES NOT have Continuous Rotation axis. Or please disable Constant Path function.
- If Continuous Rotation axis is set in robot finale axis, Constant Path function is disabled forcedly.
- If Constant Path function setup is changed by Continuous Rotation setup, original Constant Path function status is restored after Continuous Rotation setup is cleared.

Using the function

After setting up the continuous rotation axis, specify the start point of continuous rotation using the operation add instruction, ”continuous rotation speed instruction”.

The following ”continuous rotation speed instruction” is supported. The ”continuous rotation speed instruction” must be specified as an operation add instruction.

* The specification method is the same as that for other operation add instructions, and is therefore omitted.

(→ Subsection 5.3.4, ”Teaching an Additional Motion Instruction” in FANUC Robot series R-30*i*B/R-30*i*B Mate CONTROLLER OPERATOR’S MANUAL (Basic Operation) (B-83284EN))

- Continuous rotation speed instruction CTV

* where i = –100 to 100, which is the ratio of the rotation axis speed to the maximum axis speed (%)

Starting continuous rotation

Continuous rotation is started as soon as an operation statement with a continuous rotation speed instruction added is started.

Stopping continuous rotation

Continuous rotation is stopped when the first operation statement with no continuous rotation speed instruction added is started since a continuous rotation speed instruction was started.

When continuous rotation is stopped, the operation on the other axes for the same operation group also terminates. The robot, therefore, decelerates even if the positioning format for the previous operation is CNT.

The robot starts decelerating to stop on the continuous rotation axis after it has completely stopped on the other axes. At this time, the robot is not necessarily at the specified position on the continuous operation axis. Thus, the synchronization of the operation on the continuous rotation axis with the operation on the other axes (including those for other operation groups) is lost.

If an operation statement is specified next, the robot rotates in the same direction as the previous continuous rotation direction to move to the specified position.

Notes

- Continuous rotation continues even if logic instructions (instructions other than those in operation statements) are executed.
- During program playback, the turn number for the continuous rotation axis is ignored, and is always assumed to be ”0”.
- The turn number for the continuous rotation axis at a point specified when this function is enabled is always stored as ”0”.
- If the rotation axis speed for a continuous rotation speed instruction is specified as ”0”, continuous operation is not performed. If an operation statement is specified next, shorter-way operation is performed on the continuous rotation axis. This feature is useful if continuous rotation about the continuous rotation axis is to be stopped temporarily but temporary stop of the robot due to the end of the continuous rotation is to be avoided. (See the next section, ”Example of use”.)
- In single-step execution (both forward and backward), continuous rotation is not performed even if a continuous rotation speed instruction is added; shorter-way operation is performed.
- Continuous rotation stops due to a hold. If program execution is subsequently restarted, if the target position has already been reached on axes other than the continuous rotation axis, continuous rotation is not performed. If the target position has not been reached on axes other than the continuous rotation axis, continuous rotation is restarted.
- Continuous rotation about the continuous rotation axis is possible from jog feed.

Example of use

The following shows an example of using the continuous rotation speed instruction.

```
1:J  P[1]   100% FINE
2:J  P[2]   100% CNT100 CTV100
3:J  P[3]   100% FINE
4:J  P[4]   100% CNT100 CTV100
5:J  P[5]   100% FINE CTV0
6:J  P[6]   100% FINE
7:J  P[7]   100% FINE CTV100
8:WAIT 100.0sec
9:J  P[8]   100% FINE
```

- Description of lines 1 to 3:
  During operation from P[1] to P[2], continuous operation is performed. Although the positioning format specified on line 2 is ”CNT”, the robot decelerates (stops temporarily on all axes at the start of the operation on line 3) because a continuous rotation speed instruction is not added to the next line, line 3.

- Description of lines 4 to 6:
  Continuous rotation starts as soon as the execution of line 4 starts. Because the rotation axis speed specified with the continuous rotation speed instruction on line 5 is 0, continuous rotation stops temporarily at the start of the execution of line 5. Because continuous rotation continues, the positioning format CNT100 on line 4 is valid and the robot does not decelerate.
  When line 6 is executed, shorter-way operation is performed on the continuous rotation axis.
- Description of lines 7 to 9:
  Continuous rotation starts at the start of operation on line 7. Continuous rotation continues during the execution of the wait instruction (logic instruction) on line 8.
  The robot stops temporarily on all axes at the start of operation on line 9, and continuous rotation stops.

Notes/restrictions

Note the following when using this function:

- When continuous rotation is to be performed on a robot axis or built-in additional axis, The X and Y components of the tool coordinate system must both be 0. (Only the Z-axis component can have a value other than 0.)
  If this condition is not satisfied, the path of linear or circular or circle arc motion cannot be guaranteed in normal motion other than continuous rotation.
- This function cannot be used together with the following functions:
  - Asynchronous addition axis speed instruction. (The synchronous additional axis speed instruction can be used.)
  - Arc sensor
  - Weaving
  - TCP speed estimation function (sealing flow rate control)
  - Cannot use Incremental instruction (INC) for Continuous rotation axis. (Can use only for non-continuous rotation axis or the continuous rotation axis is disabled case.)
- This function automatically updates the mastering data (for the continuous rotation axis only) according to the amount of rotation about the continuous rotation axis. Thus, previously recorded mastering data may not match the current mastering data.
  After this function is disabled, it is not necessary to perform mastering.
- When this function is disabled, the current position on the continuous rotation axis may fall outside the stroke limits. If this occurs, move the position on continuous rotation axis within the stroke limits using jog feed or a program.
- If, on a multi-group system, the settings on the SETUP Continuous Turn screen are changed and the F4, DONE key is pressed, it is necessary to set system variable $PARAM_GROUP[group].$SV_OFF_ENB[i] (where i is an axis number) to FALSE to disable break control for all the axes for all operation groups before turning the power back on with a cold start.
- On a multi-group system, even if there are multiple continuous rotation axes, separate continuous rotation speeds cannot be specified for them.
- At the end of continuous rotation, one or more rotations about the continuous rotation may be performed to ensure smooth deceleration and stop. (The amount of rotation differs depending on the acceleration/deceleration constant.)
- Even during backward execution (single-step execution), shorter-way operation is performed on the continuous rotation axis. If, therefore, forward step execution and backward execution are performed sequentially in an operation statement with the movement angle being very close to 180o, rotation may be performed about the continuous rotation axis in the same direction during the forward and backward executions.
- Original Path Resume feature is disabled when continuous turn loaded. Even if system variable ($SCR.$ORG_PTH_RSM) set enabled, feature is disabled when cycle power.
- Groups which have Continuous Rotation axis will disable Cycle Time Priority feature for Cartesian motion. This will be longer cycle time and path change. When setup of Continuous Rotation axis is cleared, Cycle Time Priority feature for Cartesian motion will be restored original state. Please execute program carefully after changing of Continuous Rotation axis setup.

5 OPERATION GROUP DO OUTPUT FUNCTION

The operation group DO output function outputs information about the operation groups that are capable of jog feed, and about the operation groups of the programs being executed/temporarily stopped, to an external device with a digital output signal (DO) or robot output signal (RO). This allows devices other than the teach pendant to recognize the currently effective operation groups, thus improving safety. This function is effective when the multi-group option is used.

Function

This function allows the allocation of two DOs (jog signal and program signal) to a single operation group. For DOs, any digital output signals or robot output signals of the robot can be used.
Each allocated DO signal turns on/off under the following conditions:

- Jog signals

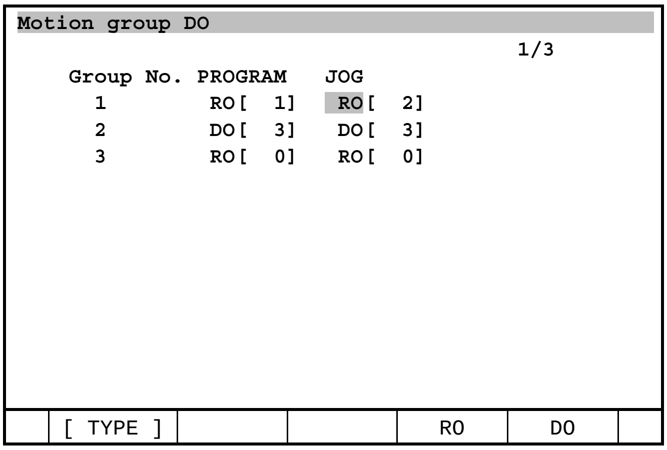
When the teach pendant is disabled, all signals turn off.
When the teach pendant is enabled, the signal for the currently selected operation group on the teach pendant turns on, while the other signals turn off.

- Program signals

Regardless of whether the teach pendant is enabled or disabled, the signal for the operation group of the program currently being executed/temporarily stopped turns on. (The signal does not turn on when the program is merely selected.)
If other programs are being executed/temporarily stopped with the multitask option, the signals for the operation groups of these programs also turn on.

Setup

To set up the operation group DO output function, use the [SETUP Motion DO] screen.
To change the signal number for an operation group, move the cursor to the signal number and enter a new value.



```
Motion group DO
                                          1/3
    Group No. PROGRAM   JOG
      1         RO[  1]  RO[  2]
      2         DO[  3]  DO[  3]
      3         RO[  0]  RO[  0]

  [ TYPE ]                      RO        DO
```

To change the type of a signal, position the cursor to the type of a signal and press function key F4, RO or F5, DO.
To disable a signal, set the number of the signal to 0.
The same signal can be set for both the program and jog signals for the same operation group. In this case, the output signal is the OR of the two signals. That is, the signal turns on if either the program or jog signal turns on. (The signal turns off only if both the program and jog signals turn off.)

Example of using this function with the multitask option

This section explains the operation of this function when a subprogram call or the multitask option is used. The output program signal is the OR of the signals for all the operation groups of the program currently being executed or temporarily stopped.

If a program without an operation group calls a program having an operation group by using a subprogram call, the signal for the operation group of the subprogram turns on only while the subprogram is being executed. (The signal does not turn on when the main program without an operation group is merely selected/executed.)

If the execution instruction of the multitask function is to start another program that operates the robot (the main program that has the execution instruction does not have an operation group), the signal for the operation group of the program started by the execution instruction does not turn on when the main program is merely selected/executed. The program signal turns on when the program that operates the robot is actually started.

Consider the following three example programs:

```
PROGRAM MAIN : Operation group[*,*,*,*,*,*,*,*]
1:RUN PRG A
2:RUN PRG B
:

PROGRAM PRG A : Operation group[1,*,*,*,*,*,*,*]
1:J  P[1]   100% FINE
:

PROGRAM PRG B : Operation group[*,1,*,*,*,*,*,*]
1:L  P[1]   500mm/sec CNT100
:
```

Program MAIN, which does not have an operation group, starts PRG A and PRG B having operation groups by using execution instructions. PRG A uses operation group 1 and PRG B uses operation group 2.

- The program signals for the groups do not turn on when program MAIN is merely selected.
- When line 1 of MAIN is executed, PRG A is started and the signal for operation group 1 turns on.
- When line 2 of MAIN is executed, PRG B is started and the signal for operation group 2 turns on.
- When PRG A and PRG B terminates, the respective signals for operation groups 1 and 2 turn off.

Notes

Note the following when using this function:

- The same signal cannot be defined for different operation groups.
- While a program is being executed/temporarily stopped, the type (DO or RO) and number of the program signal cannot be changed.

6 AUTOMATIC ERROR RECOVERY FUNCTION

To use this function, automatic error recovery function option (A05B-2xxx-J924) is required.

6.1 AUTOMATIC ERROR RECOVERY FUNCTION

Robots are sometimes stopped by various alarms even during production. If a robot is stopped, it is necessary to perform recovery operation then resume the program that was originally running. For example, suppose that a robot is performing arc welding. An alarm due to an arc start failure may be issued, stopping the robot. In such a case, the operator must jog the robot to a safe position to, for example, cut the end of the wire or clean the nozzle, then resume the original program.
The automatic error recovery function is provided to support automatic operation of the above sequence.

The automatic error recovery function consists of the following two functions.

- Resume program function
- Fast exit/entry feature

6.2 RESUME PROGRAM FUNCTION

This function is a function to remove the cause of the alarm by using the resume program when the robot is stopped by the alarm when being producing, and to resume the original program.

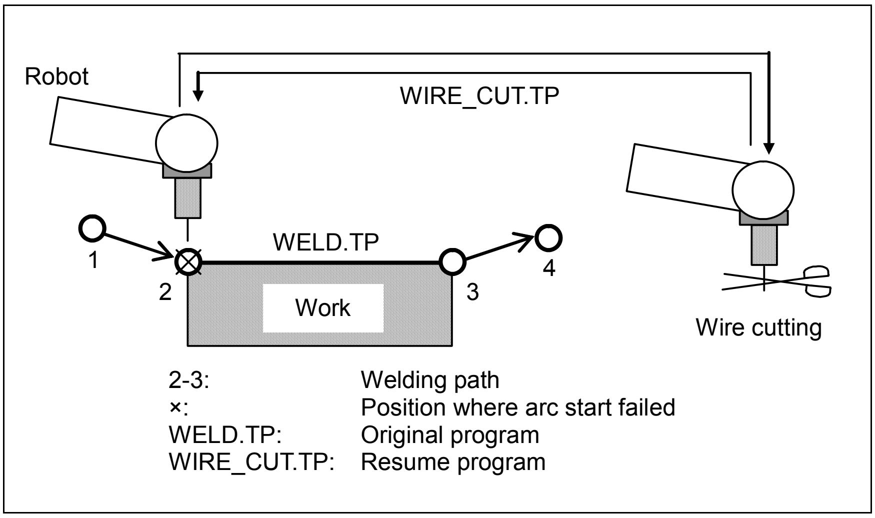

Robot
WIRE_CUT.TP
WELD.TP
1
2
3
4
Work
Wire cutting
2-3: Welding path
×: Position where arc start failed
WELD.TP: Original program
WIRE_CUT.TP: Resume program

Fig.6.2 Resume program function

In the example shown above, the robot is operated by executing WELD.TP to perform welding along the path from 2 to 3.
Assume that an arc start failure occurs at the arc start position 2. At this time, if the resume program function is used, another program called the resume program, which is WIRE_CUT.TP in this case, can be started at the next start signal input. After this program terminates, another start signal input resumes the original program. If the resume operation function is then enabled (which is set on the welding system setting screen), the robot automatically returns to the original position where the robot was stopped, then the original program is resumed. If the return distance for resume operation is set, the robot returns from the stop position by the set distance, then the original program is resumed. If no arc is produced, a scratch start takes place.

6.3 FAST EXIT/ENTRY FEATURE

If an alarm is issued during operation in a complicated environment, the robot moves from the stopped position to the taught point to execute the resume program. In this case, the robot may interfere with part of a workpiece or peripheral devices. After recovery operation, similar interference may occur when an attempt is made to execute the original program. The fast exit/entry feature is provided to avoid the possibility of such interference.

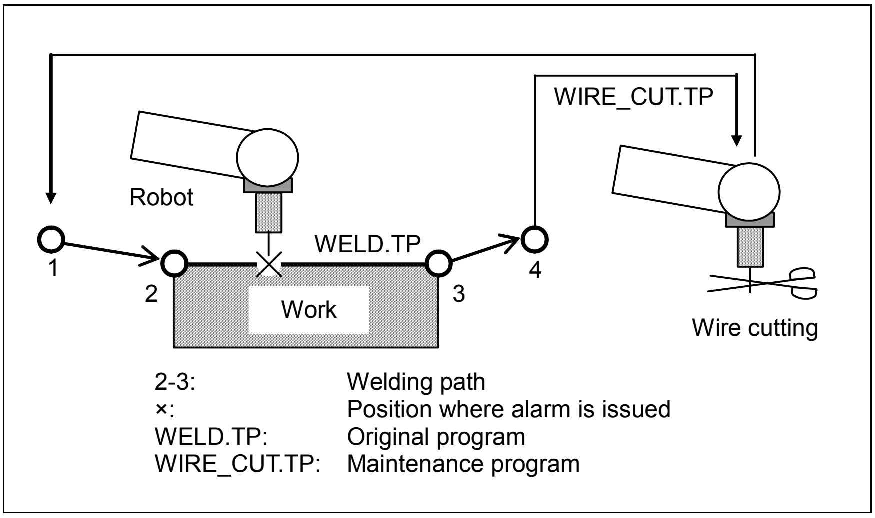

WIRE_CUT.TP
Robot
WELD.TP
1
2
3
4
Work
Wire cutting
2-3: Welding path
×: Position where alarm is issued
WELD.TP: Original program
WIRE_CUT.TP: Maintenance program

Fig.6.3 Fast exit/entry feature

In the example shown above, the robot is operated by executing WELD.TP to perform welding along the path from 2 to 3. At this time, when the alarm of something is issued between positions 2-3 and the robot stops, the fast exit/entry feature automatically does the following operation.

1. From the stopped position, disable arc welding, and execute only the move statements of the original program up to the end.
2. Execute a maintenance program, which is WIRE_CUT.TP in this case.
3. Disable arc welding, execute the move statements of the original program from the beginning to move the robot to the stopped position.
4. Enable arc welding, and resume the original program operation.

Thereafter, in this manual, above-mentioned operation 1-4 is called FFR sequence (Fast Fault Recovery sequence).

6.4 RESUME_PROG INSTRUCTION

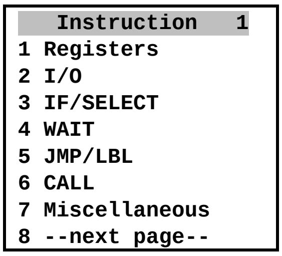
The resume program function executes a resume program defined in an original program, in lieu of the original program. The resume program instructions to define the resume program are the following five instructions. RESUME_PROG[1] can be used in single-task/multi-task, and RESUME_PROG[2-5] can be used in the multi-task.

- RESUME_PROG[1:Comment] : For Single-task/Multi-task
- RESUME_PROG[2:Comment] : For Multi-task
- RESUME_PROG[3:Comment] : For Multi-task
- RESUME_PROG[4:Comment] : For Multi-task
- RESUME_PROG[5:Comment] : For Multi-task

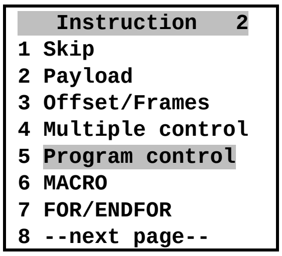
To erase the defined resume program, use the following instruction.

- CLEAR_RESUME_PROG

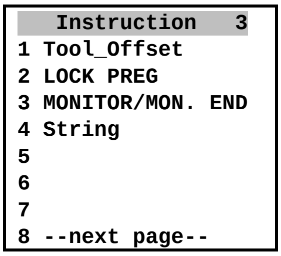
To teach the resume instructions, press F1, [INST] to display the related menu, then select “Program control” from the menu.



```
  Instruction   1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Miscellaneous
8 --next page--
```



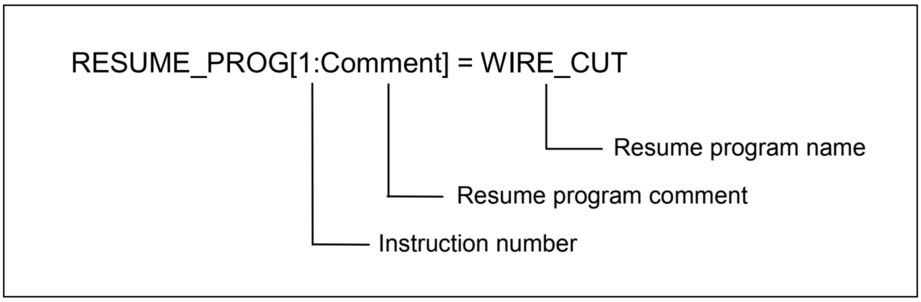
```
  Instruction   2
1 Skip
2 Payload
3 Offset/Frames
4 Multiple control
5 Program control
6 MACRO
7 FOR/ENDFOR
8 --next page--
```



```
  Instruction   3
1 Tool_Offset
2 LOCK PREG
3 MONITOR/MON. END
4 String
5
6
7
8 --next page--
```

RESUME_PROG[1:Comment]

The program name defined by RESUME_PROG[1:Comment] is registered as a resume program when RESUME_PROG[1:Comment] is executed in original program, and when the alarm is issued while defining the resume program, the resume program is executed.



To use the resume program instructions, the following option is needed.

- To use the RESUME_PROG[1-5], the Automatic error recovery function (A05B-2600-J924) is needed.
- In addition, to use the RESUME_PROG[2-5], the following options are needed.

RESUME_PROG[2]

Dual Arm Control	A05B – 2600 – J605

RESUME_PROG[2-5]

Multi UOP interface function	A05B – 2600 – J964

When the instruction number of RESUME_PROG[ ] is defined, it is necessary to accord the following conditions.

TP Program that can be started by START#1/PROD_STAR#1 of UOP	RESUME_PROG[1:Comment]
TP Program that can be started by START#2/PROD_STAR#2 of UOP	RESUME_PROG[2:Comment]
TP Program that can be started by START#3/PROD_STAR#3 of UOP	RESUME_PROG[3:Comment]
TP Program that can be started by START#4/PROD_STAR#4 of UOP	RESUME_PROG[4:Comment]
TP Program that can be started by START#5/PROD_STAR#5 of UOP	RESUME_PROG[5:Comment]

For instance, in multi-task, the RESUME_PROG[3:Comment] can be used only by the program that can be started by START#3/ PROD_STAR#3 of UOP.
The RESUME_PROG[2-5] cannot be used in the single-task because it is for multi-task. Therefore, use the RESUME_PROG[1:Comment] in the single-task.

To define the resume program instructions, there are the following conditions.

- Multiple resume program instructions cannot be defined in the original program.
- The resume program instructions cannot be defined in the resume/maintenance program.
- A resume program instruction the same instruction number cannot be executed by the plural. For instance, the RESUME_PROG[5] cannot be executed by other programs while the resume program is already being defined by the RESUME_PROG[5]. However, the RESUME_PROG[1-4] can be executed by other programs.
- The resume program instructions cannot be defined in the programs executed by the RUN instruction.

CLEAR_RESUME_PROG

When the CLEAR_RESUME_PROG is executed, the resume program defined by RESUME_PROG[ ] is erased. Therefore, even if the alarm is issued after the CLEAR_RESUME_PROG is executed, the resume program is not executed. Moreover, the CLEAR_RESUME_PROG is effective only in the program in which the CLEAR_RESUME_PROG is defined. For instance, in multi task, when the CLEAR_RESUME_PROG is executed in the program in which the RESUME_PROG[1] is defined, only the resume program defined by the RESUME_PROG[1] is erased. At this time, the resume program defined by the RESUME_PROG[2-5] is never erased.

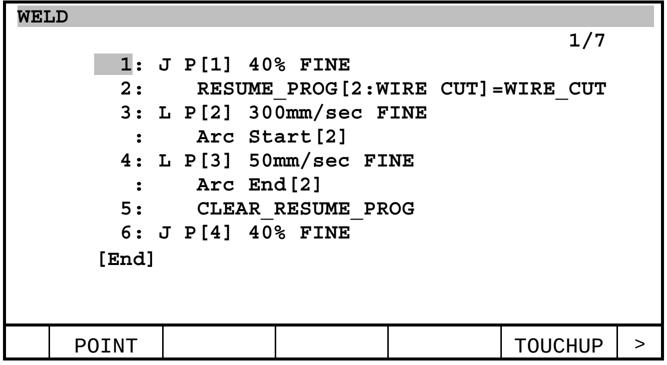
The resume program is erased also when:

- Backward execution is performed in original program.
- The cursor line is changed manually in original program.
- The original program terminates.

> ⚠ **CAUTION**
> When the resume program is erased while the resume program is pausing, the resume program cannot be restarted. The original program restarts from the pause position of the resume program.

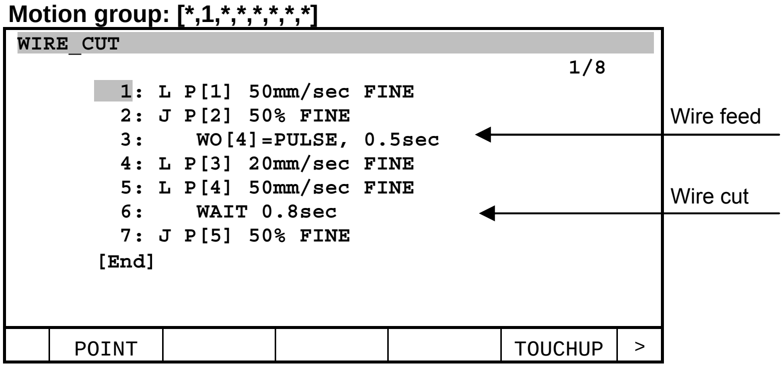
The following programs are examples that use the RESUME_PROG[2] and the CLEAR_RESUME_PROG.

WELD.TP
Motion group: [*,1,*,*,*,*,*,*]



```
WELD
                                             1/7
      1: J P[1] 40% FINE
      2:    RESUME_PROG[2:WIRE CUT]=WIRE_CUT
      3: L P[2] 300mm/sec FINE
       :    Arc Start[2]
      4: L P[3] 50mm/sec FINE
       :    Arc End[2]
      5:    CLEAR_RESUME_PROG
      6: J P[4] 40% FINE
    [End]
```

WIRE_CUT.TP
Motion group: [*,1,*,*,*,*,*,*]

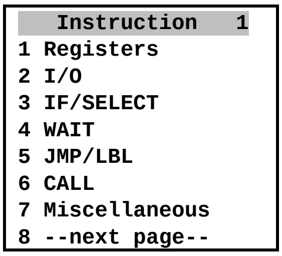


```
WIRE_CUT
                                                1/8
       1: L P[1] 50mm/sec FINE
       2: J P[2] 50% FINE
       3:    WO[4]=PULSE, 0.5sec
       4: L P[3] 20mm/sec FINE
       5: L P[4] 50mm/sec FINE
       6:    WAIT 0.8sec
       7: J P[5] 50% FINE
      [End]
```

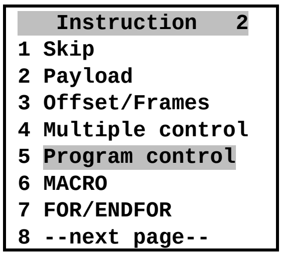
In the above program example, the WIRE_CUT.TP is taught in the second line of the WELD.TP and is erased in the fifth line. Since the WIRE_CUT.TP is defined as the resume program between the third to fourth lines, it is executed as the resume program. In the sixth and subsequent lines, the resume program has been erased, so the resume program is not executed.

6.5 RETURN_PATH_DSBL INSTRUCTION

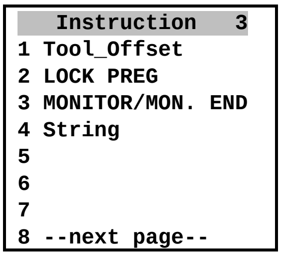
In arc tool systems, the resume operation function is generally enabled. With this function enabled, return to the original stop position is always performed then arc is produced when the original program resumes after the resume program terminates. In some systems, however, return to the original stop position should not sometimes be performed. For example, when the nozzle touch state is input through DI, a resume program is used to relieve the torch slightly in the torch direction. If the resume operation function operates, return to the original stop position is performed even when relieve operation has been performed. As a result, the nozzle touch state is observed again. In such a case, the resume operation function needs to be kept enabled, but it should be disabled only after the execution of the resume program.
This can be performed with the RETURN_PATH_DSBL instruction. By using this instruction within the resume program, the resume operation function can be disabled only when the original program is resumed next. This instruction is valid only when it is executed within a resume program; the instruction is invalid when executed in a program other than the resume program.
The RETURN_PATH_DSBL instruction appears in the menu containing the RESUME_PROG[] instruction.



```
  Instruction   1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Miscellaneous
8 --next page--
```



```
  Instruction   2
1 Skip
2 Payload
3 Offset/Frames
4 Multiple control
5 Program control
6 MACRO
7 FOR/ENDFOR
8 --next page--
```



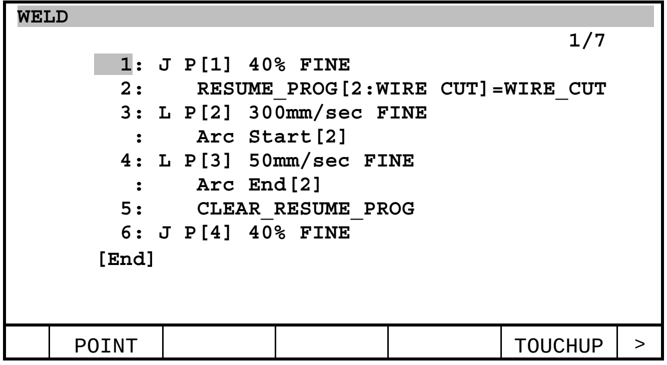
```
  Instruction   3
1 Tool_Offset
2 LOCK PREG
3 MONITOR/MON. END
4 String
5
6
7
8 --next page--
```

RETURN_PATH_DSBL

The RETURN_PATH_DSBL is effective only in the resume program in which the RETURN_PATH_DSBL is defined. For instance, in multi task, when RETURN_PATH_DSBL is executed in the resume program defined by RESUME_PROG[1], only the resume operation function of the original program in which RESUME_PROG[1] is defined becomes invalid. The resume operation function of the original program in which other RESUME_PROG[2-5] is defined is effective.

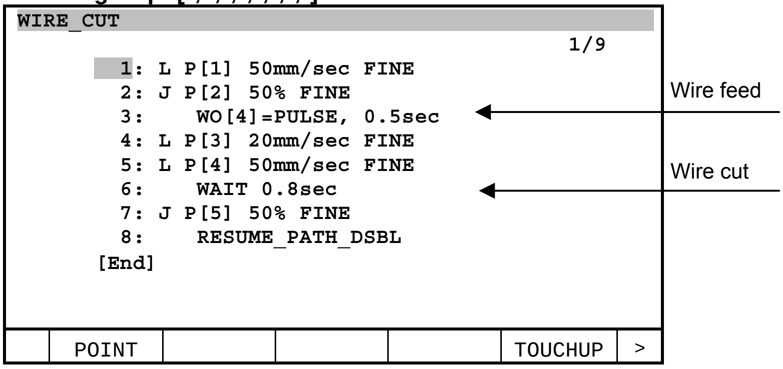
The following programs are examples that use the RETURN_PATH_DSBL. The RETURN_PATH_DSBL instruction is valid only when it is taught within resume program. Use this instruction as shown in the sample program given below. If the instruction is taught as shown below, the resume operation function does not operate when the original program resumes after the resume program terminates, even if the resume operation function is enabled.

WELD.TP
Motion group: [*,1,*,*,*,*,*,*]



```
WELD
                                                  1/7
       1: J P[1] 40% FINE
       2:    RESUME_PROG[2:WIRE CUT]=WIRE_CUT
       3: L P[2] 300mm/sec FINE
        :    Arc Start[2]
       4: L P[3] 50mm/sec FINE
        :    Arc End[2]
       5:    CLEAR_RESUME_PROG
       6: J P[4] 40% FINE
      [End]
```

WIRE_CUT.TP
Motion group: [*,1,*,*,*,*,*,*]



```
WIRE_CUT
                                                  1/9
       1: L P[1] 50mm/sec FINE
       2: J P[2] 50% FINE
       3:    WO[4]=PULSE, 0.5sec
       4: L P[3] 20mm/sec FINE
       5: L P[4] 50mm/sec FINE
       6:    WAIT 0.8sec
       7: J P[5] 50% FINE
       8:    RESUME_PATH_DSBL
      [End]
```

6.6 MAINT_PROG INSTRUCTION

The fast exit/entry feature automatically does the following operation.

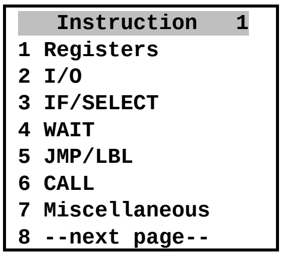
1 From the stopped position, disable arc welding, and execute only the move statements of the original program up to the end.
2 Execute a maintenance program.
3 Disable arc welding, execute the move statements of the original program from the beginning to move the robot to the stopped position.
4 Enable arc welding, and resume the original program operation.

Thereafter, in this manual, above-mentioned operation 1-4 is called FFR sequence (Fast Fault Recovery sequence).

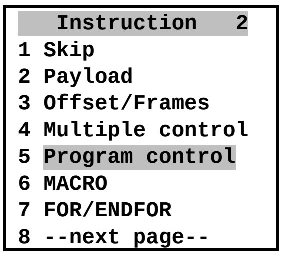
The maintenance program instructions to define the maintenance program are the following five instructions. MAINT_PROG[1] can be used in single-task/multi-task, and MAINT_PROG[2-5] can be used in the multi-task.

• MAINT_PROG[1:Comment] :	For Single-task/Multi-task
• MAINT_PROG[2:Comment] :	For Multi-task
• MAINT_PROG[3:Comment] :	For Multi-task
• MAINT_PROG[4:Comment] :	For Multi-task
• MAINT_PROG[5:Comment] :	For Multi-task

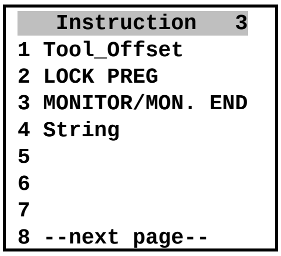
The MAINT_PROG[ ] instruction appears in the menu containing the RESUME_PROG[ ] instruction.



```
  Instruction   1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Miscellaneous
8 --next page--
```

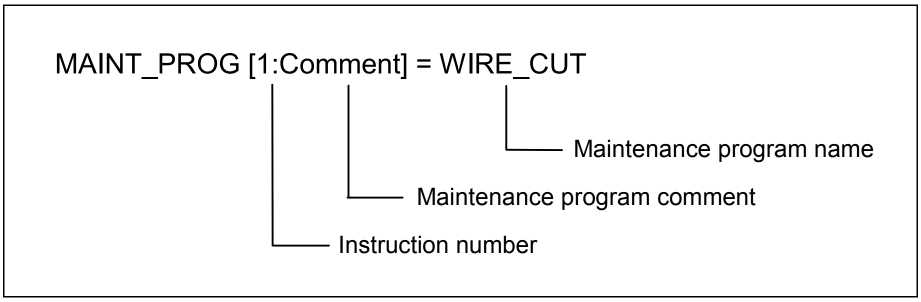


```
  Instruction   2
1 Skip
2 Payload
3 Offset/Frames
4 Multiple control
5 Program control
6 MACRO
7 FOR/ENDFOR
8 --next page--
```



```
  Instruction   3
1 Tool_Offset
2 LOCK PREG
3 MONITOR/MON. END
4 String
5
6
7
8 --next page--
```

MAINT_PROG[1:Comment]


MAINT_PROG [1:Comment] = WIRE_CUT
Maintenance program name
Maintenance program comment
Instruction number

The program name defined by MAINT_PROG[1:Comment] is registered as a maintenance program when MAINT_PROG[1:Comment] is executed in original program, and when the alarm is issued while defining the maintenance program, the FFR sequence is executed.

To use the maintenance program instructions, the following option is needed.

- To use the MAINT_PROG[1-5], the Automatic error recovery function (A05B-2600-J924) is needed.
- In addition, to use the MAINT_PROG[2-5], the following options are needed.

MAINT_PROG[2]

Dual Arm Control	A05B – 2600 – J605

MAINT_PROG[2-5]

Multi UOP interface function	A05B – 2600 – J964

When the instruction number of MAINT_PROG[ ] is defined, it is necessary to accord the following conditions.

TP Program that can be started by START#1/PROD_STAR#1 of UOP	MAINT_PROG[1:Comment]
TP Program that can be started by START#2/PROD_STAR#2 of UOP	MAINT_PROG[2:Comment]
TP Program that can be started by START#3/PROD_STAR#3 of UOP	MAINT_PROG[3:Comment]
TP Program that can be started by START#4/PROD_STAR#4 of UOP	MAINT_PROG[4:Comment]
TP Program that can be started by START#5/PROD_STAR#5 of UOP	MAINT_PROG[5:Comment]

For instance, in multi-task, the MAINT_PROG[3:Comment] can be used only by the program that can be started by START#3/ PROD_STAR#3 of UOP.
The MAINT_PROG[2-5] cannot be used in the single-task because it is for multi-task. Therefore, use the MAINT_PROG[1:Comment] in the single-task.

When the FFR sequence is completed, the following messages are displayed on the teach pendant.

MOTN-431 Fast fault recovery 1	When the FFR sequence executed by the MAINT_PROG[1] is completed.
MOTN -432 Fast fault recovery 2	When the FFR sequence executed by the MAINT_PROG[2] is completed.

MOTN -433 Fast fault recovery 3	When the FFR sequence executed by the MAINT_PROG[3] is completed.
MOTN -434 Fast fault recovery 4	When the FFR sequence executed by the MAINT_PROG[4] is completed.
MOTN -435 Fast fault recovery 5	When the FFR sequence executed by the MAINT_PROG[5] is completed.

To define the maintenance program instructions, there are the following conditions.

- Multiple maintenance program instructions cannot be defined in the original program.
- The maintenance program instructions cannot be defined in the resume/maintenance program.
- A maintenance program instruction the same instruction number cannot be executed by the plural. For instance, the MAINT_PROG[5] cannot be executed by other Programs while the maintenance program is already being defined by the MAINT_PROG[5]. However, the MAINT_PROG[1-4] can be executed by other Programs.
- The maintenance program instructions cannot be defined in the programs executed by the RUN instruction.

The maintenance program is erased also when:

- Backward execution is performed in original program or maintenance program.
- The cursor line is changed manually in original program or maintenance program.
- The original program or the maintenance program terminates.

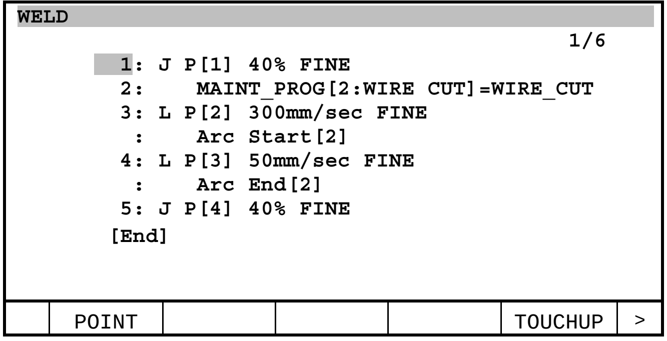
⚠ CAUTION

When the definition of the maintenance program is deleted while executing the FFR sequence, the maintenance program or the original program cannot be restarted by the FFR sequence.

The following programs are examples that use the MAINT_PROG[2].

WELD.TP

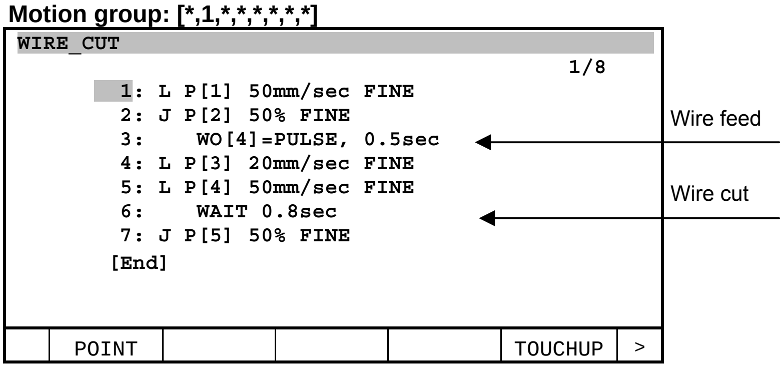
Motion group: [*,1,*,*,*,*,*,*]



```
WELD
                                             1/6
      1: J P[1] 40% FINE
      2:    MAINT_PROG[2:WIRE CUT]=WIRE_CUT
      3: L P[2] 300mm/sec FINE
       :    Arc Start[2]
      4: L P[3] 50mm/sec FINE
       :    Arc End[2]
      5: J P[4] 40% FINE
     [End]
```

WIRE_CUT.TP

Motion group: [*,1,*,*,*,*,*,*]



```
WIRE_CUT
                                             1/8
      1: L P[1] 50mm/sec FINE
      2: J P[2] 50% FINE
      3:    WO[4]=PULSE, 0.5sec
      4: L P[3] 20mm/sec FINE
      5: L P[4] 50mm/sec FINE
      6:    WAIT 0.8sec
      7: J P[5] 50% FINE
     [End]
```

In the above program example, the WIRE_CUT.TP is taught in the second line of the WELD.TP. Since the WIRE_CUT.TP is defined as the maintenance program between the third to final lines, it is executed as the maintenance program. And when the execution of the WELD.TP ends, the definition of the maintenance program is automatically deleted.

6.7 SETTING OF THE AUTOMATIC ERROR RECOVERY FUNCTION

On the setting screen of the automatic error recovery function, the following settings can be made:

- Instruction number : Selecting the instruction number

Error recovery function common setup

- Error Recovery Function : Enabling/disabling the automatic error recovery function
- Approval DI Index No.* : Defining the recovery switch DI
- Incomplete End DO Index No.* : Defining the incomplete end DO
- Reset DI Index No.* : Defining the incomplete-end reset DI
- Automatic Start Feature : Enabling/disabling the alarm-time automatic start feature
- Alarm setting screen : Enabling/disabling the alarm code monitoring function
- DI_alarm setting screen : Defining the automatic error recovery alarm conditions

Resume program type recovery

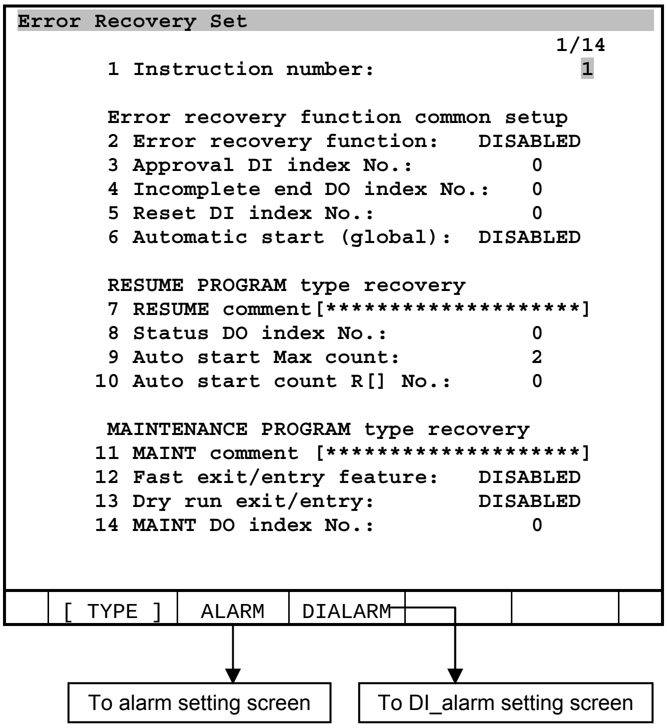
- RESUME comment : Defining the resume program comment
- Status DO Index No.* : Defining the resume program information DO
- Auto Start Max Count : Setting the maximum number of automatic start repetitions
- Auto Start Max Count R[ ] : Setting the automatic start count register

Maintenance program type recovery

- MAINT comment : Defining the maintenance program comment
- Fast Exit/Entry Feature : Enabling/disabling the fast exit/entry feature
- Dry Run Exit/Entry : Enabling/disabling dry run exit/entry operation
- MAINT DO Index No.* : Defining the FFR sequence information DO

(* Set this to zero if you do not want to use this feature.)

To display the setting screen of the automatic error recovery function, first press [MENU] key to display the screen menu, then select ”6 SETUP”. Then, press F1, [TYPE] to display the screen switching menu, then select Err recovery.
[MENU] → 6 SETUP → Err recovery



```
Error Recovery Set
                                          1/14
       1 Instruction number:                1

       Error recovery function common setup
       2 Error recovery function:   DISABLED
       3 Approval DI index No.:         0
       4 Incomplete end DO index No.:   0
       5 Reset DI index No.:            0
       6 Automatic start (global):  DISABLED

       RESUME PROGRAM type recovery
       7 RESUME comment[********************]
       8 Status DO index No.:           0
       9 Auto start Max count:          2
      10 Auto start count R[] No.:      0

       MAINTENANCE PROGRAM type recovery
      11 MAINT comment [********************]
      12 Fast exit/entry feature:   DISABLED
      13 Dry run exit/entry:        DISABLED
      14 MAINT DO index No.:            0
```

Selecting the instruction number

This screen can be displayed by each instruction number of the resume program instruction and the maintenance program instruction. It is possible to switch to a setup screen of other instruction numbers by changing this item.

For multi-task

For instance, instruction number 1-3 can be selected by this item in the system that can use RESUME_PROG[1-3] and MAINT_PROG[1-3].

Error recovery function common setup

Enabling/disabling the automatic error recovery function

This item enables or disables the automatic error recovery function. When the automatic error recovery function is enabled, and neither monitored alarm codes nor the recovery switch DI are defined, the resume program is always executed at restart from the suspended state (except when the resume program information DO is off). And, when the automatic error recovery function and the fast exit/entry feature are enabled, and neither monitored alarm codes nor the recovery switch DI are defined, the FFR sequence is always executed at restart from the suspended state(except when the FFR sequence information DO is off). When this item is disabled, the resume program and the FFR sequence is not executed.

For multi-task

This item can be set by each instruction number.

Defining the recovery switch DI

To use the recovery switch DI function, define a DI number. After the number is defined, power must be turned off then back on.

With this function, the operator can choose whether to execute the resume program or the FFR sequence or not at the time of restart from the suspended state by using a peripheral device. When DI is OFF, neither the resume program nor the FFR sequence are executed. If this signal number is not defined, this function is disabled.

The specifications of the recovery switch DI function are listed below.

Table 6.7 (a) Specifications of the recovery switch DI function

DI number definition	Recovery switch DI function status	DI status	Execution of resume program or FFR sequence at restart
0	Disabled	□□	Executed
Valid number defined	Enabled	On	Executed
		Off	Not executed

⚠ **CAUTION**

To continue a resume program at program restart after the resume program is suspended, input the on state of the recovery switch DI. If it is off, the original program is executed.

For multi-task

This item can be set by each instruction number.

Defining the incomplete end DO

When an incomplete end DO number is defined, the incomplete end DO is output if a certain forced termination alarm is issued during execution of the resume program or the maintenance program. The output incomplete end DO is turned off by the next start signal input.

Before inputting the start signal, the operator must check the incomplete end DO signal status. If this signal is on, the resume program or the maintenance program terminates in the middle, so the robot is not in a specified position.

⚠ **CAUTION**

When the resume program or the maintenance program terminates in the middle, and the start signal is input, the robot might do action not predictable. Therefore, before inputting the start signal, check the program name and the program start line that will be executed with the next start signal. And, if an interfering object exists, jog the robot to a position where interfering object doesn't exist, then input the start signal.

This signal may be added to the PLC start signal acceptance conditions.
If this signal is set to 0, this function is disabled.

For multi-task

This item can be set by each instruction number.

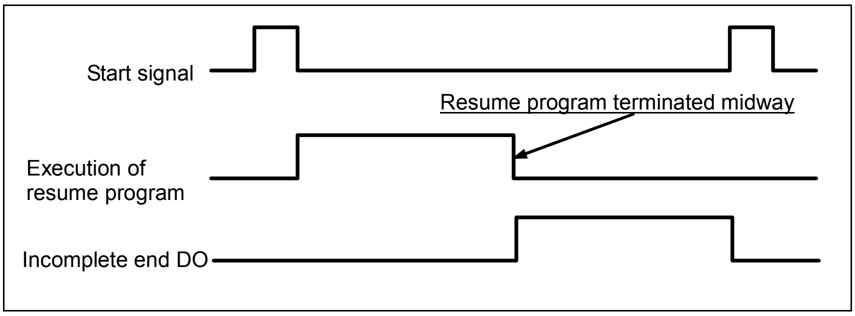


Fig. 6.7 (a) Incomplete end DO output timing chart

Defining the incomplete-end reset DI

When the incomplete end DO is included in the PLC start signal acceptance conditions, the operator requires a means to turn off the incomplete end DO externally. Inputting the incomplete-end reset DI signal

turns off the incomplete end DO. When the incomplete end DO is output, the operator must first check the program name and the program start line that will be executed with the next start signal. And, if an interfering object exists, jog the robot to a position where interfering object doesn't exist, input the incomplete-end reset DI, then input the start signal.
If this signal is set to 0, this function is disabled.

For multi-task

This item can be set by each instruction number.

Enabling/disabling the alarm-time automatic start feature

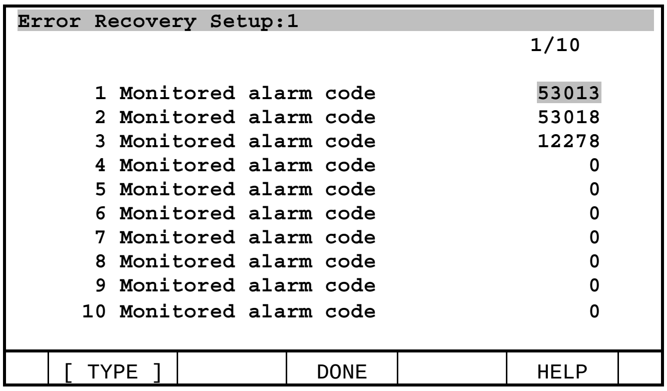
When this item is enabled, and an alarm defined in the monitored alarm codes is issued, the resume program or the FFR sequence are automatically executed without outputting the alarm signal and stopping the robot. When the resume program has terminated, the original program is resumed automatically by this function (In the fast exit/entry feature, when the maintenance program has terminated, the original program is resumed automatically by FFR sequence.). During then, the operator need not input the start signal.
Because the alarm signal is not output, other robots are not stopped when multiple robots are operating. The robot for which the alarm was issued moves by itself to the recovery station, and after recovery work, the original program is resumed.
When this item is disabled, and an alarm defined in the monitored alarm codes is issued, the program outputs an alarm signal and stops running. Input of the start signal executes the defined resume program or FFR sequence. In the resume program function, after the resume program terminates, another start signal input restarts the suspended original program. Another start signal to resume an original program need not be input for the fast exit/entry feature. The original program is resumed automatically by FFR sequence.

For multi-task

- This item cannot be set by each instruction number. When this item is enabled in either of instruction number, this function of all the instruction numbers are automatically enabled.
- Even if this function is enabled, this function is not executed when the alarm defined in the monitored alarm codes is issued while the alarm is being issued in other robots. In that case, the program outputs an alarm signal and stops running. Input of the start signal executes the defined resume program or FFR sequence.

Enabling/disabling the alarm code monitoring function

When this function is enabled, and only when an alarm defined in the monitored alarm codes is issued, the resume program or the FFR sequence is executed. When the monitored alarm code is defined, and the resume program or the maintenance program is defined, this function is enabled automatically. If no alarm code is defined, this function is disabled. In this case, before the suspended original program is resumed, the resume program or the FFR sequence is always executed (except when the recovery switch DI function is enabled).
To define alarm codes to be monitored, press the F2, ALARM key. A screen for defining alarm codes is displayed.



```
Error Recovery Setup:1
                                              1/10

      1 Monitored alarm code             53013
      2 Monitored alarm code             53018
      3 Monitored alarm code             12278
      4 Monitored alarm code                 0
      5 Monitored alarm code                 0
      6 Monitored alarm code                 0
      7 Monitored alarm code                 0
      8 Monitored alarm code                 0
      9 Monitored alarm code                 0
     10 Monitored alarm code                 0
```

When a defined alarm code is issued, and a program is suspended, the resume program or FFR sequence are executed at restart.
Each alarm code consists of an alarm code ID and alarm number. The alarm code ID indicates the type of alarm. For an arc start failure alarm, for example, the following alarm code is indicated:

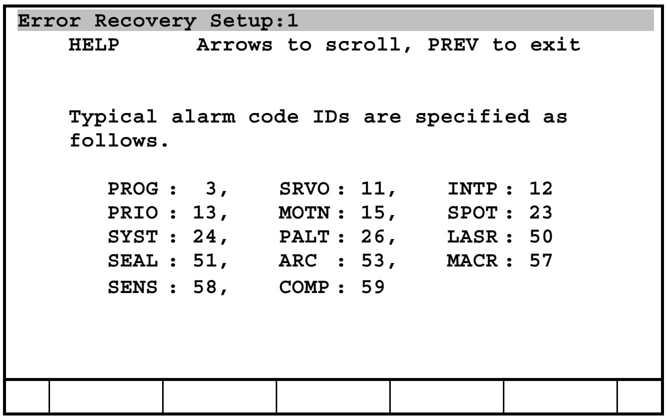
ARC – 013 Arc Start failed = 53 013
ID (53) Number ID Number

For alarm numbers, refer to FANUC Robot series R-30*i*B/R-30*i*B Mate CONTROLLER OPERATOR'S MANUAL (Alarm Code List) (B-83284EN-1).

Up to ten alarm codes can be defined as standard. To change the maximum number of alarm codes (up to 20 codes) that can be defined, change system variable $RSMPRG_SV.$NUM_ALARM, cycle power off the controller.

Instruction number	System variable
1	$RSMPRG_SV.$NUM_ALARM
2	$RSMPRG_SV2.$NUM_ALARM
3	$RSMPRG_SV3.$NUM_ALARM
4	$RSMPRG_SV4.$NUM_ALARM
5	$RSMPRG_SV5.$NUM_ALARM

Pressing the F5, HELP key displays the following screen:



```
Error Recovery Setup:1
    HELP      Arrows to scroll, PREV to exit


    Typical alarm code IDs are specified as
    follows.

       PROG :  3,    SRVO : 11,    INTP : 12
       PRIO : 13,    MOTN : 15,    SPOT : 23
       SYST : 24,    PALT : 26,    LASR : 50
       SEAL : 51,    ARC  : 53,    MACR : 57
       SENS : 58,    COMP : 59
```

If the following conditions are met, the alarm code monitoring function is disabled:

- The recovery switch DI is off.
- There is no alarm code defined.
- The resume program or the maintenance program is not defined in the program currently being executed.

- The resume program or the FFR sequence is executing.

The specifications of the alarm code monitoring function are listed below.

Table 6.7 (b) Specifications of the alarm code monitoring function

Alarm code definition	Alarm code function status	Issuance of defined alarm	Execution of resume program or FFR sequence at restart
0	Disabled	□□	Executed
At least one alarm code is defined	Enabled	Issued	Executed
		Not issued	Not executed

> ⚠ **CAUTION**
> Defined alarms must have the suspension alarm attribute. Do not define any warning alarm as an alarm code.

> **For multi-task**
> This item can be set by each instruction number.

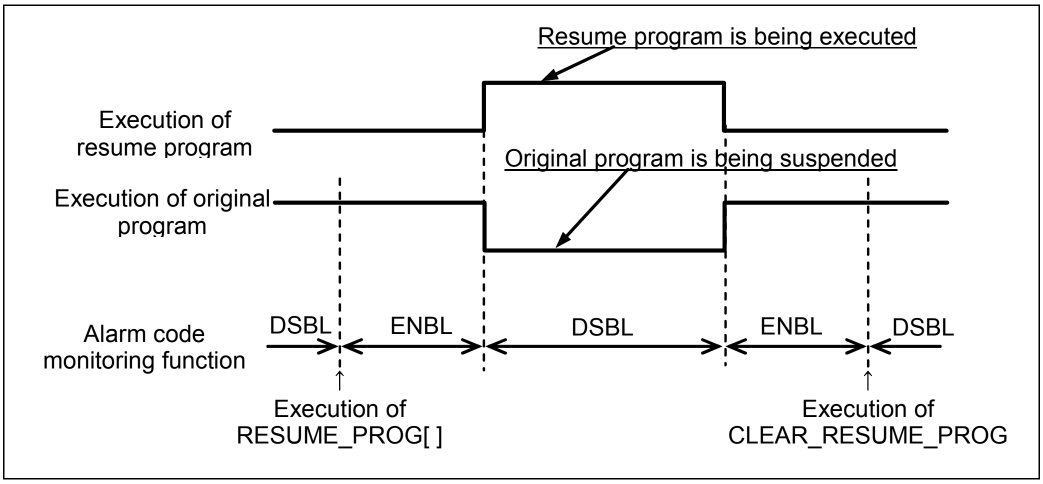

Resume program is being executed
Execution of resume program
Original program is being suspended
Execution of original program
Alarm code monitoring function
DSBL
ENBL
DSBL
ENBL
DSBL
Execution of RESUME_PROG[ ]
Execution of CLEAR_RESUME_PROG

Fig. 6.7 (b) Enabling/disabling the alarm code monitoring function in the resume program function

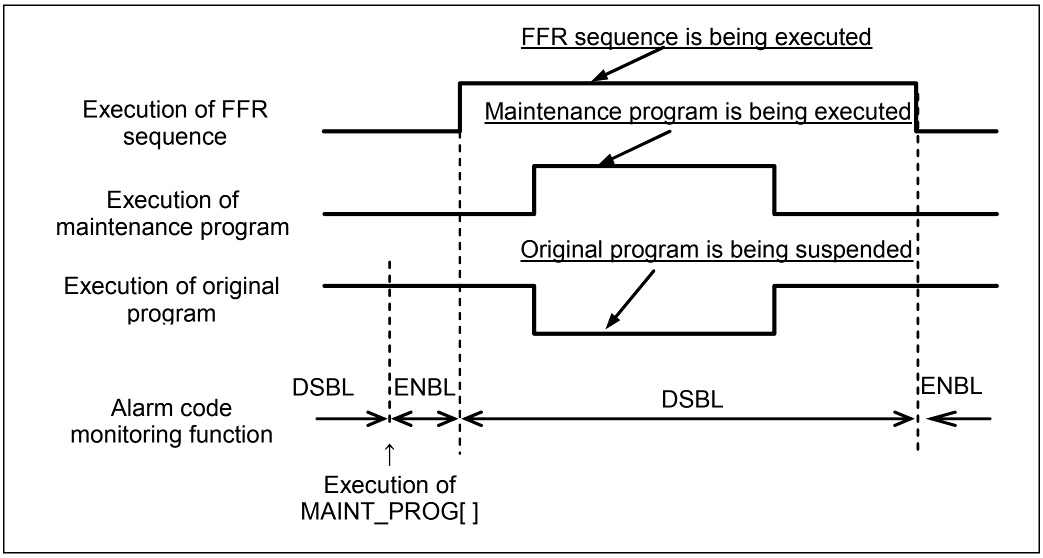

FFR sequence is being executed
Execution of FFR sequence
Maintenance program is being executed
Execution of maintenance program
Original program is being suspended
Execution of original program
Alarm code monitoring function
DSBL
ENBL
DSBL
ENBL
Execution of MAINT_PROG[ ]

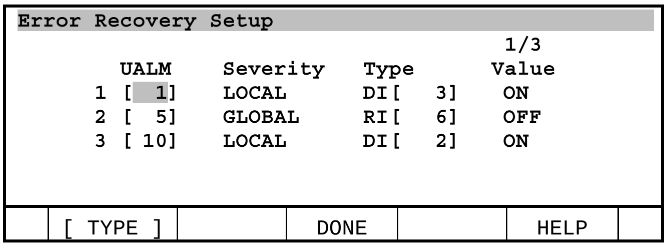
Fig. 6.7 (c) Enabling/disabling the alarm code monitoring function in the fast exit/entry feature

Defining automatic error recovery alarm conditions

By inputting a defined digital input signal, an automatic error recovery alarm can be issued. When this alarm is defined for the alarm-time automatic start feature, the resume program or the FFR sequence can be executed automatically by inputting the digital input signal.

As the message for an automatic error recovery alarm, a message defined for a user alarm can be used. The alarm severity can be set to either LOCAL or GLOBAL selectively. When LOCAL is selected, the alarm is issued only for a program that defines the resume program or the maintenance program. The status of a digital input signal to be monitored can be set by selecting the signal type from among DI, RI, and WI, changing the signal number, and selecting the trigger status between on and off.

Define the conditions for issuing an automatic error recovery alarm on the definition screen that is displayed by pressing F3, DI_ALARM.



```
Error Recovery Setup
                                          1/3
       UALM     Severity    Type       Value
     1 [  1]    LOCAL       DI[   3]    ON
     2 [  5]    GLOBAL      RI[   6]    OFF
     3 [ 10]    LOCAL       DI[   2]    ON
```

On this screen, the items shown below can be set. The alarm code of the automatic error recovery alarm is 12278.

- User alarm number
  When the automatic error recovery alarm is issued, the user alarm message with the set number is displayed as an alarm message. When this item setting has been changed, the new setting becomes effective immediately.
- Alarm severity
  This item can choose whether the automatic error recovery alarm is a local alarm or global alarm. When LOCAL is set, the automatic error recovery alarm is issued only for the program that defines the resume program or the maintenance program. If there is no program that defines the resume program or the maintenance program, the alarm is regarded as a global alarm. When this item setting has been changed, the new setting becomes effective immediately.
- Signal type
  Choose the type of the digital signal for issuing the automatic error recovery alarm from among DI, RI, and WI. When this item setting has been changed, the power must be turned off then back on for the new setting to become effective.
- Signal number
  Set the number of the digital signal for issuing the automatic error recovery alarm. When this setting has been changed, the power must be turned off then back on for the new setting to become effective.
- Detection signal status
  Set the status of the digital signal for issuing the automatic error recovery alarm to ON (high) or OFF (low). When this setting has been changed, the power must be turned off then back on for the new setting to become effective.

The standard number of automatic error recovery alarm conditions is three. This number can be increased to up to five by changing system variable $RSMPRG_SV.$NUM_DI_ALM. After this system variable has been changed, the power must be turned off then back on for the new setting to become effective.

For multi-task

- This item cannot be set by each instruction number. When this item is enabled in either of instruction number, this function of all the instruction numbers are automatically enabled.
- When the alarm severity issues the automatic error recovery alarm of LOCAL alarm when there are multiple programs that define the resume program or the maintenance program, the automatic error recovery alarm of the number of programs is issued.
- Define the automatic error recovery alarm code in all the instruction numbers that can be used when the automatic start feature is enabled, and the automatic error recovery alarm code is defined in the monitored alarm codes. When LOCAL alarm is defined in the automatic error recovery alarm only by the limited instruction number, even if the LOCAL alarm is issued while multiple programs are executed at the same time, the automatic start feature might not be able to be executed.

Resume program type recovery

Defining the resume program comment

A comment can be specified for the resume program instruction. Move the cursor to this item and press the [ENTER] key. The screen enters comment input mode. A resume program comment is used to describe additional information to be displayed on the program edit screen together with the resume program instruction.

For multi-task

This item can be set by each instruction number.

Defining the resume program information DO

When the alarm code monitoring function and recovery switch DI function are both disabled, the resume program is always executed at the time of restart after the original program is suspended. When both the functions are enabled, it is difficult to determine whether the original program or resume program is to be executed at restart.

The resume program information DO is on only when the resume program is executed at restart. When the signal is off, the original program is executed at restart. With this function, the operator can know which program is to be executed next.

If the following conditions are met, the resume program information DO goes on:

- The automatic error recovery function is enabled.
- Not the single step mode.
- The resume program is defined in the original program.
- The original program has a motion group.
- The original program is suspended, and the resume program is not yet completed.
- There is no optional function that disables the automatic error recovery function. See “Other specifications and restrictions”.
- The user condition parameter is true. See “Conditions for executing the resume program”.

RESUME_PROG instruction	User condition parameter
RESUME_PROG[1:Comment]	$AUTORCV_ENB
RESUME_PROG[2:Comment]	$AUTORCVENB2
RESUME_PROG[3:Comment]	$AUTORCVENB3
RESUME_PROG[4:Comment]	$AUTORCVENB4
RESUME_PROG[5:Comment]	$AUTORCVENB5

- When the teach pendant is enabled:

  - The operation mode (on the automatic error recovery manual operation screen) is TP_TEST.
- When the teach pendant is disabled:
  - The operation mode (on the automatic error recovery manual operation screen) is AUTO.
  - The remote conditions are met when system variable $RMT_MASTER is 0.
  - There is no alarm code defined. If any alarm code is defined, the alarm code is issued.
  - The recovery switch DI function is disabled. If this function is enabled, the recovery switch DI signal is on.

⚠ **CAUTION**

1. While the resume program is being executed, single step operation cannot be performed.
2. Even if the resume program information DO is on, the resume program is not executed when backward execution of the original program is performed.
3. Backward execution in the resume program is possible.
4. The update cycle period for the resume program information DO is 300 ms. When the conditions listed above have been changed, wait 300 ms before program execution.

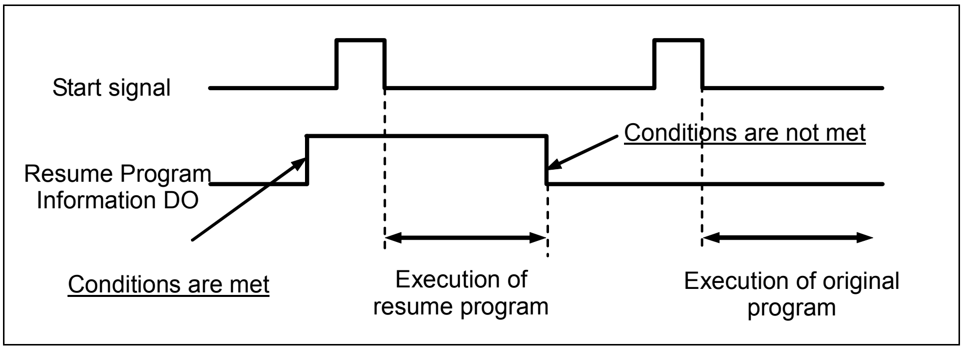

Start signal
Resume Program Information DO
Conditions are not met
Conditions are met
Execution of resume program
Execution of original program

Fig. 6.7 (d) Resume program information DO output timing chart

For multi-task
This item can be set by each instruction number.

Setting the maximum number of automatic start repetitions

When a defined alarm is issued, the alarm-time automatic start feature automatically executes the resume program, then resumes the original program. If the defined alarm is issued again when the original program is resumed, the automatic start feature functions again. For example, the automatic start feature is activated by an alarm indicating an arc start failure, then the same alarm is issued again when the original program has resumed.

To prevent such an endlessly repeated condition, set the maximum number of automatic start repetitions. The number of times the resume program is started repeatedly is counted internally. If the count exceeds the set value, “INTP-134 Over automatic start Max counter” is issued, and the resume program information DO is turned off at the same time. If this occurs, eliminate the cause of the alarm issued in the original program. Then input the start signal.

⚠ **CAUTION**

The number of repetitions counted internally is cleared when the execution of a move statement has terminated and when the CLEAR_RESUME_PROG instruction has been executed.

For multi-task
This item can be set by each instruction number.

Defining the automatic start count register

As mentioned above, the resume program may be executed several times repeatedly by the automatic start feature. When the automatic start count register is defined, a different program can be executed as the resume program each time the resume program is executed. For example, when the resume program is executed for the first time by the automatic start feature, the register value is 1. When an alarm is issued again during execution of the original program, and the resume program is then executed again by the automatic start feature, the register value is 2. By executing a different subprogram in the resume program according to the register value, different resume program operation can be performed each time the repetition count is incremented.

> ⚠ **CAUTION**
> When the resume program is executed by other than the automatic start feature, the register value is 0. Therefore, a resume program must be created so that the same subprogram is called when the register value is 0 and when the value is 1.

> **For multi-task**
> This item can be set by each instruction number.

Maintenance program type recovery

Defining the maintenance program comment

A comment can be specified for the maintenance program instruction. Move the cursor to this item and press [ENTER] key. The screen enters comment input mode. A maintenance program comment is used to describe additional information to be displayed on the program edit screen together with the maintenance program instruction.

> **For multi-task**
> This item can be set by each instruction number.

Enabling/disabling the fast exit/entry feature

This item enables or disables the fast exit/entry feature.

When the automatic error recovery function and the fast exit/entry feature are enabled, and neither monitored alarm codes nor the recovery switch DI are defined, the FFR sequence is always executed at restart from the suspended state (except when the FFR sequence information DO is off).

When this item is disabled, the FFR sequence is not executed.

> **For multi-task**
> This item can be set by each instruction number.

Enabling/disabling dry run exit/entry operation

In the fast exit/entry feature, this item specifies whether exit from the stopped position and return to the stopped position after maintenance program execution are to be performed at dry run speed. This item can be set only for instruction number 1 in the single-task.

> **For multi-task**
> This item can be set by each instruction number.

Defining the FFR sequence information DO

When the alarm code monitoring function and recovery switch DI function are both disabled, the FFR sequence is always executed at the time of restart after the original program is suspended. When both the functions are enabled, it is difficult to determine whether the original program or the FFR sequence is to be executed at restart.

The FFR sequence information DO is on only when the FFR sequence is executed at restart. And, because the FFR sequence is not executed though the original program is executed at restart when the signal is off, the maintenance program is not executed after the original program is completed. With this function, the operator can know the FFR sequence is to be executed next.

If the following conditions are met, the FFR sequence information DO goes on:

- The automatic error recovery function is enabled.
- Not the single step mode.
- The maintenance program is defined in the original program.
- The resume program information DO is not on.
- The FFR sequence is being executed.
- The fast exit/entry feature is enabled.
- When the teach pendant is disabled:
    - The remote conditions are met when system variable $RMT_MASTER is 0.
    - There is no alarm code defined. If any alarm code is defined, the alarm code is issued.
    - The recovery switch DI function is disabled. If this function is enabled, the recovery switch DI signal is on.

⚠ **CAUTION**

1 Perform neither backward execution nor the single step operation while the FFR sequence is executed.

2 The update cycle period for the FFR sequence information DO is 300 ms. When the conditions listed above have been changed, wait 300 ms before program execution.

3 When the teach pendant is enabled, the FFR sequence is not executed.

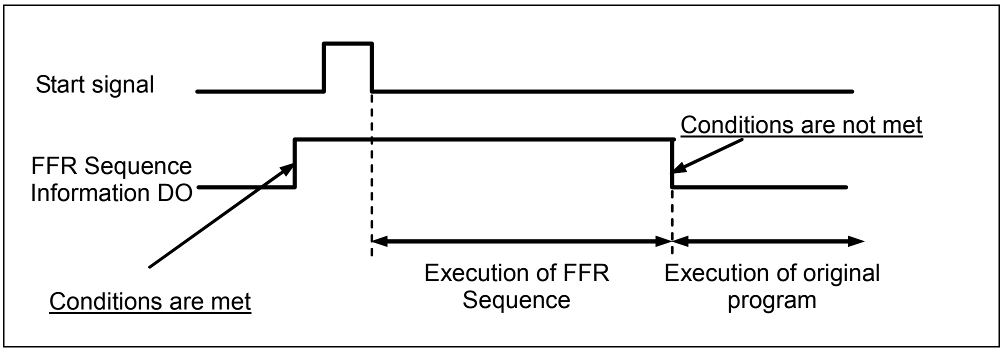

Start signal
FFR Sequence Information DO
Conditions are not met
Conditions are met
Execution of FFR Sequence
Execution of original program

Fig. 6.7 (e) FFR sequence information DO output timing chart

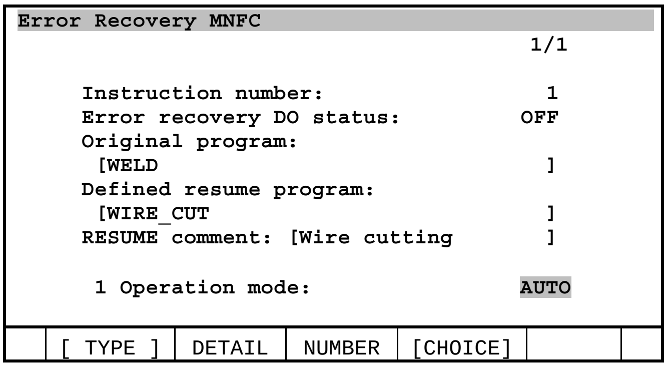
For multi-task

This item can be set by each instruction number.

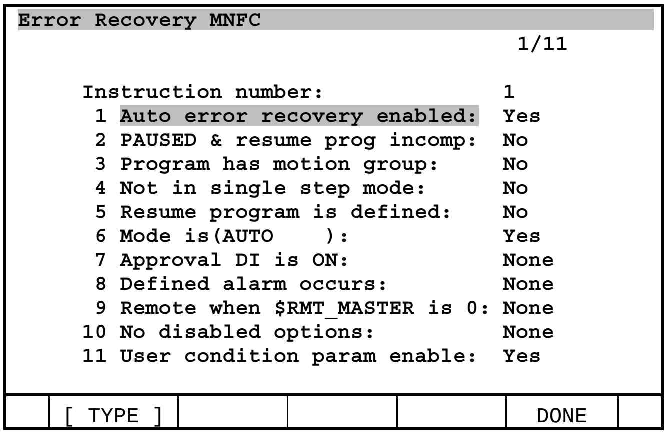
6.8 MANUAL OPERATION SCREEN OF THE RESUME PROGRAM FUNCTION

A manual operation screen is supported for the resume program function (The fast exit/entry feature is not supported.). To display the manual operation screen, first press MENU to display the screen menu, then select ”3 MANUAL FCTNS”. Then, press F1, [TYPE] to display the screen switching menu, then select Err recovery.

[MENU] → 3 MANUAL FCTNS → Err recovery



```
Error Recovery MNFC
                                           1/1

     Instruction number:                     1
     Error recovery DO status:             OFF
     Original program:
      [WELD                                  ]
     Defined resume program:
      [WIRE_CUT                              ]
     RESUME comment: [Wire cutting           ]

      1 Operation mode:                   AUTO

   [ TYPE ]  DETAIL   NUMBER  [CHOICE]
```



```
Error Recovery MNFC
                                          1/11

     Instruction number:                  1
      1 Auto error recovery enabled:      Yes
      2 PAUSED & resume prog incomp:      No
      3 Program has motion group:         No
      4 Not in single step mode:          No
      5 Resume program is defined:        No
      6 Mode is(AUTO    ):                Yes
      7 Approval DI is ON:                None
      8 Defined alarm occurs:             None
      9 Remote when $RMT_MASTER is 0:     None
     10 No disabled options:              None
     11 User condition param enable:      Yes

   [ TYPE ]                               DONE
```

Selecting the instruction number
The instruction number of the resume program instruction is indicated. It is possible to switch to a manual operation screen of other instruction numbers by changing this item.

Resume program information DO status
The resume program information DO status is indicated. Even when the resume program information DO is not defined, its status can be indicated. From this information, the operator can know which program, the resume program or original program, is to be executed.

Original program
The name of the original program that defines the resume program is indicated.

Defined resume program
The name of the resume program defined in the currently selected program is indicated. From this information, the operator can check whether a wrong resume program is defined or not.

> ⚠ **CAUTION**
> If a wrong program is defined as the resume program, the robot operation is unpredictable. Therefore, check that the resume program is correct.

Resume program comment
The comment specified for the resume instruction is indicated.

Operation mode
There are three operation modes. The standard setting is AUTO. When the display changes from this screen to another, AUTO is automatically set again.

Operation mode	Explanation
AUTO	This mode should be set when the teach pendant is disabled. When this mode is selected, the resume program is executed according to the status of the alarm code monitoring function and recovery switch DI function. If this mode is selected when the teach pendant is enabled, the resume program is not executed.
NO_EXEC	When this mode is selected, the resume program information DO is always off. Therefore, in this mode, the resume program is not executed.
TP_TEST	This mode should be set when the teach pendant is enabled. When this mode is selected, and when the teach pendant is enabled, the resume program is always executed regardless of the status of the alarm code monitoring function or error recovery switch DI function.

Displaying detail conditions of the resume program information DO

When F2, DETAIL is pressed on the manual operation screen of the automatic error recovery function, detail conditions related to the resume program information DO status are displayed. When all items on the detail screen are set to Yes or None, the resume program information DO is turned on. When the resume program information DO is off, and you cannot find the cause of the DO being off, check this screen.

Auto error recovery enabled
This item indicates whether this function is enabled or disabled on the setting screen of the automatic error recovery function.

PAUSED & resume prog incomp
This item indicates the following conditions:

- The original program must be in the suspended state.
- A resume program must be defined in the original program, and the execution of the resume program must not have been completed.

Program has motion group
This item indicates that the selected program has a motion group.

Not in single step mode
This item indicates that the single step mode is not set.
The single step LED on the teach pendant indicates the single step status of the original program. Even when the single step key is pressed while the resume program is suspended, and the single step LED goes on, the resume program information DO is held on. This is because the LED indicates that the original program is in single step mode; the resume program is not in single step mode.

Resume program is defined
This item indicates that a resume program is defined in the original program.

Mode is (xxxx)
This item indicates that the operation mode is suitable for the current status.
For example, when the teach pendant is disabled, “AUTO” is indicated in the portion “xxxx.” When the teach pendant is enabled, “TP_TEST” is indicated.

Approval DI is ON
This item indicates the recovery switch DI status. When the DI number is not defined, or when the teach pendant is enabled, “None” is indicated.

Defined alarm occurs
This item indicates that an alarm code is defined, and that alarm is issued.
When no alarm code is defined, or when the teach pendant is enabled, “None” is indicated.

Remote when $RMT_MASTER is 0
This item indicates that remote conditions are met. This function is enabled only when the teach pendant is disabled, system variable $RMT_MASTER is 0, and system variable $RSMPRG_SV.$CHK_REMOTE is true.

No disabled options
There are options that cannot be used together with the automatic error recovery function. This item indicates whether such options are present or not.

User condition param enable
This item indicates the status of the user condition parameter. For how to use the user condition parameter, see “Changing conditions for executing the resume program”.

6.9 EXECUTION OF THE RESUME PROGRAM FROM THE TEACH PENDANT AND TEST MODE

Normally, the automatic error recovery function is used when production is started with the teach pendant disabled. When checking the resume program during teaching, set the operation mode to TP_TEST on the manual operation screen. In TP_TEST mode, the resume program can be executed regardless of the recovery switch DI status and whether a defined alarm is issued or not. A manual operation screen is not supported for the fast exit/entry feature.

6.10 CHANGING CONDITIONS FOR EXECUTING THE RESUME PROGRAM

To use resume program execution conditions other than alarm codes, use user condition parameter and the status monitoring function.

RESUME_PROG instruction	User condition parameter
RESUME_PROG[1:Comment]	$AUTORCV_ENB
RESUME_PROG[2:Comment]	$AUTORCVENB2
RESUME_PROG[3:Comment]	$AUTORCVENB3
RESUME_PROG[4:Comment]	$AUTORCVENB4
RESUME_PROG[5:Comment]	$AUTORCVENB5

For example, to execute the resume program when R[1] is 1, create the following monitor program, and start MONIT1.CH on the system monitor screen.

```
MONIT1.CH
1:WHEN R[1]=1,CALL DORESUME
2:WHEN R[1]<>1,CALL NORESUME
```

```
DORESUME.TP
1:$AUTORCV_ENB=1
2:MONITOR MONIT3
```

```
MONIT2.CH
1:WHEN R[1]=1,CALL DORESUME
```

```
NORESUME.TP
1:$AUTORCV_ENB=0
2:MONITOR MONIT2
```

```
MONIT3.CH
1:WHEN R[1]<>1,CALL NORESUME
```

The start conditions can be changed by modifying the monitor program. For how to use the status monitoring function, refer to the Status monitoring function section in Utility chapter of R-30*i*B/R-30*i*B

Mate Controller operator’s manual (Basic Operation) (B-83284EN). In this case, the automatic start function is unavailable.

6.11 OTHER SPECIFICATIONS AND RESTRICTIONS

Automatic error recovery function

- The automatic error recovery function supports the power failure handling function.
- The automatic error recovery function is disabled when one of the following options is loaded:
  - Arc sensor
  - AVC (TIG arc length control)
  - Root path memorization
  - Line tracking
  - Soft float
  - Continuous turn
  - Remote TCP

Resume program function

- While the resume program is being executed, single step operation is not performed. Single step mode is valid only for the original program.
- When the cursor line is changed and executed while the original program is suspended, the resume program is not executed.
- While the resume program is being executed, the resume program execution status cannot be checked on the program edit screen. On the edit screen, the suspended original program is displayed.
- The resume program instructions cannot be defined in the program executed by the RUN instruction.
- When the program execution is done after the cursor line in the original program is moved while the resume program is suspended, the popup menu to confirm the movement of the cursor is displayed. When Yes is selected for the question on the popup menu, the original program is executed starting from the new cursor line. When No is selected, the resume program is restarted.
- Programs other than the original program cannot be selected by PNSTROBE signal while the resume program is executed.
- When PNSTROBE signal has been on when the resume program starts, the alarm is issued. Turn off PNSTROBE signal when the resume program starts.
- When programs other than the original program are selected while executing an original program, the original program is automatically selected again at the same time as the resume program's starting/ending.
- Never teach the arc and weaving instructions in the resume program. If an arc instruction is executed in the resume program while arc welding is being performed by the original program, alarm is issued. In addition, weaving operation is not performed within the resume program.

Fast exit/entry feature

- When an original program is a single step mode, the FFR sequence is not started.
- When the cursor line is changed and executed while the original program is suspended, the FFR sequence is not executed.
- The maintenance program instructions cannot be defined in the program executed by the RUN instruction.
- Programs other than the maintenance program cannot be selected by PNSTROBE signal while the FFR sequence is executed.
- When PNSTROBE signal has been on when the FFR sequence starts/ends, the alarm is issued. Turn off PNSTROBE signal when the FFR sequence starts/ends.
- When programs other than the original program are selected while executing an original program, the original program is automatically selected again at the same time as the FFR sequence 's starting.

6.12 WARNINGS (Be sure to read this section for safety.)

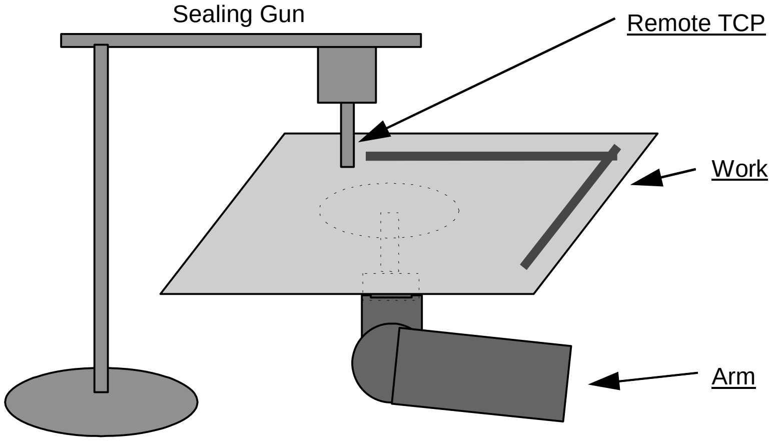
When using the automatic error recovery function, observe the following safety precautions:

- If a wrong program or a program causing wrong operation is defined as a resume program or a maintenance program, the robot moves in a direction the operator cannot predict. Define a correct program.
- Before inputting the start signal and before pressing [FWD] key on the teach pendant, for safety, check the resume program information DO status to confirm whether the original program or resume program is to be started.
- Before inputting the start signal and before pressing [FWD] key on the teach pendant, for safety, check the FFR sequence information DO status to confirm whether the original program or maintenance program is to be started.
- If the operation mode is set to TP-TEST on the manual operation screen of the automatic error recovery function, the resume program is started even when a defined alarm is not issued or when the recovery switch DI is off.
- When an operation mode other than AUTO is set on the manual operation screen of the automatic error recovery function, then the display is changed to another screen, the operation mode is set to AUTO again automatically. To use an operation mode other than AUTO, always keep displaying the manual operation screen of the automatic error recovery function.
- When the resume program or the maintenance program terminates in the middle, and the start signal is input, the robot might do action not predictable. Therefore, before inputting the start signal, check the program name and the program start line that will be executed with the next start signal. And, if an interfering object exists, jog the robot to a position where interfering object doesn't exist, then input the start signal.
- Perform neither backward execution nor the single step operation while the FFR sequence is executed.

7 REMOTE TCP FUNCTION

The REMOTE TCP function is used to process the work by moving the work which is placed on the robot hand. Robot can keep the relation between the tool fixed on the ground and work.

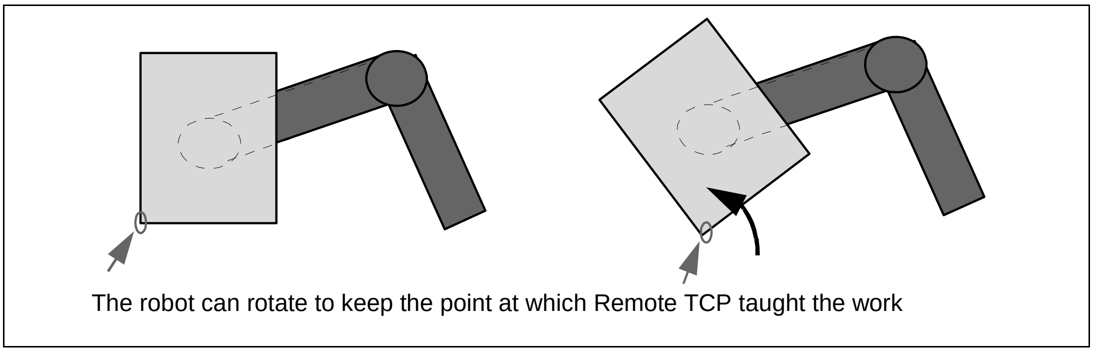
Example of using REMOTE TCP function (by Sealing application)



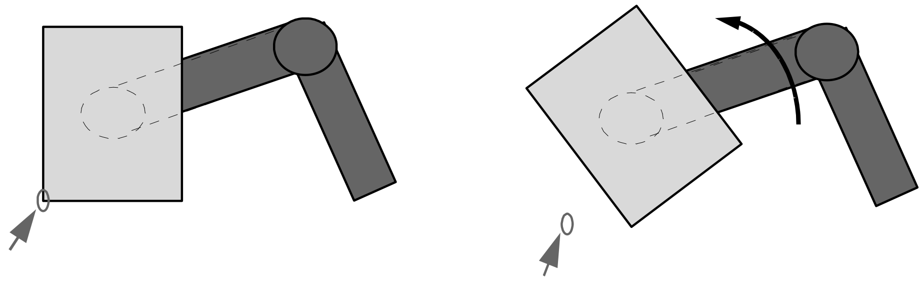
Advantage of Remote TCP

- The gun is fixed on the ground, then the cabling of tool is easy.
- It is not necessary to take the heavy gun.
- Even if you do not use the REMOTE TCP function, you could process the work by moving the work which is placed on the robot hand. But if you rotate the work against the TCP, the tool can not do the coordinate motion against the work.

Case of using REMOTE TCP function

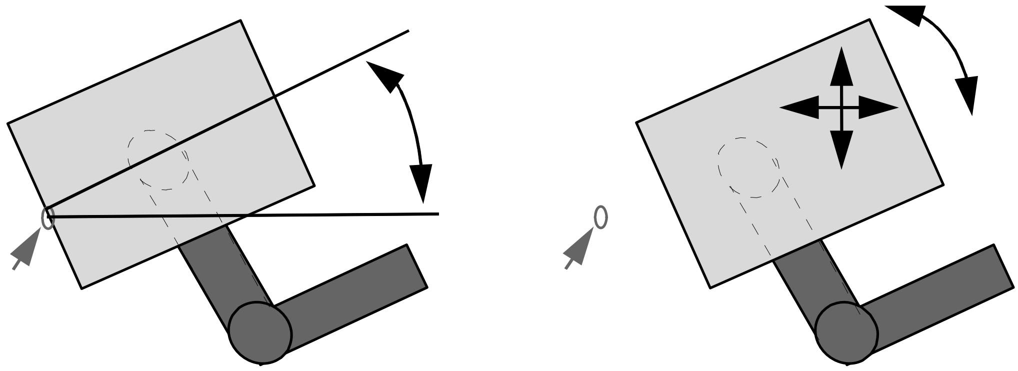

The robot can rotate to keep the point at which Remote TCP taught the work

Case of NOT using REMOTE TCP function



As the center of rotation is changed by the variation of work position, the robot can not rotate to keep the point at which Remote TCP taught the work.

- You can get the easy teaching operation by remote TCP jog function.



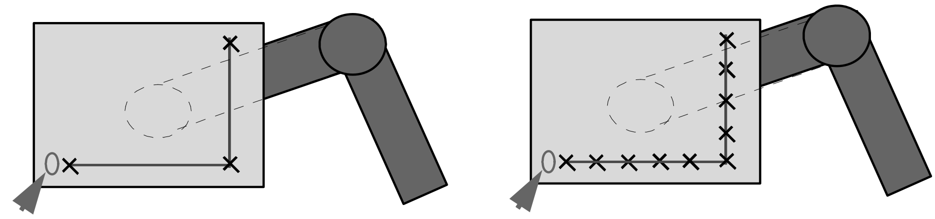
Remote TCP

Easy to jog the Remote TCP regardless of location and orientation.

No Remote TCP

No easy to jog the Remote TCP.

- You can reduce teaching point drastically.



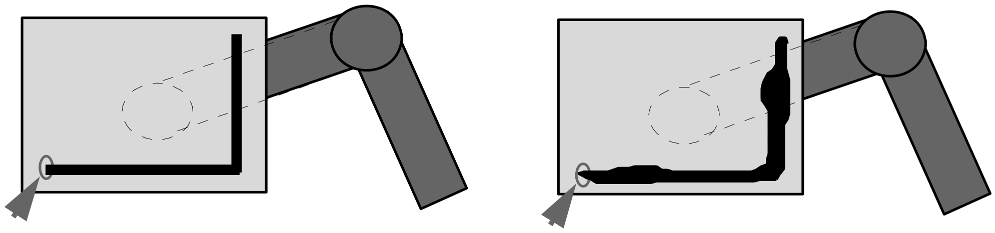
Remote TCP

You need the minimum teaching points regardless of the orientation and location of work.

No Remote TCP

You need many teaching points according to the direction and position of work.

- You can do uniform sealing against work.



Remote TCP

The robot moves with the taught speed, so you can do uniform sealing.

No Remote TCP

You can not do teaching to keep the relative speed, so you can not do uniform sealing.

Required option

- Remote TCP (A05B-2600-J624)

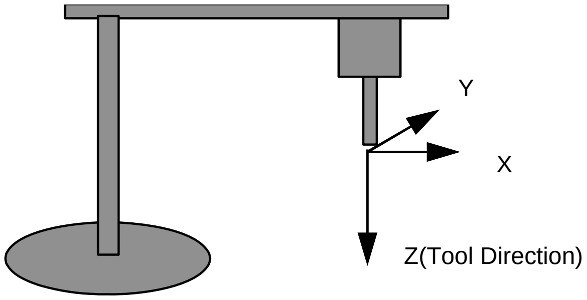
Limitation

- REMOTE TCP cannot be used in joint motion and Wrist joint motion.
- REMOTE TCP cannot be used in the robot which is installed the coordinated motion function.
- REMOTE TCP cannot be used with the following functions.
  - Line tracking
  - Weaving
  - Touch sensor
  - Arc sensor
  - AVC(TIG arc length control)
  - Root path memorization
  - Singularity avoidance
- In TCP speed output function, when the robot motion changes from REMOTE TCP to normal, the accuracy of the speed prediction function may lower.

7.1 SETUP

Setting up

If you use the REMOTE TCP function, you need to teach the tool center point which is fixed on the ground to the robot.



You can set REMOTE TCP position by using the same method for User frame.

1 Press [MENU] key.

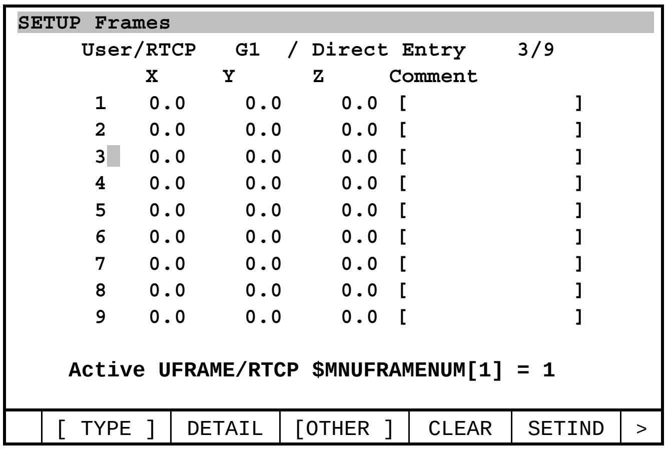
2 Select SETUP.

3 Press F1, [TYPE].

4 Select Frames.

5 If user frame is not displayed, press F3, [OTHER], and select User/RTCP If F3, [OTHER], is not displayed, press PREV.

6 Move the cursor to the REMOTE TCP frame to use.



```
SETUP Frames
     User/RTCP   G1  / Direct Entry    3/9
          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 UFRAME/RTCP $MNUFRAMENUM[1] = 1

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

7 Press F2, DETAIL.

8 Select the method from Direct Entry, Three Point or Four Point, and set the frame by using the same method for User frame.

Remote TCP jog

When you do transitional jogging under the remote TCP mode, the robot behaves the same as it is under normal jogging.

REMOTE TCP jogging is done by using the following procedure.

1 Enter to REMOTE TCP jogging mode.

2 Select the frame.

JOINT, REMOTE TCP USER, REMOTE TCP TOOL, REMOTE TCP JOG

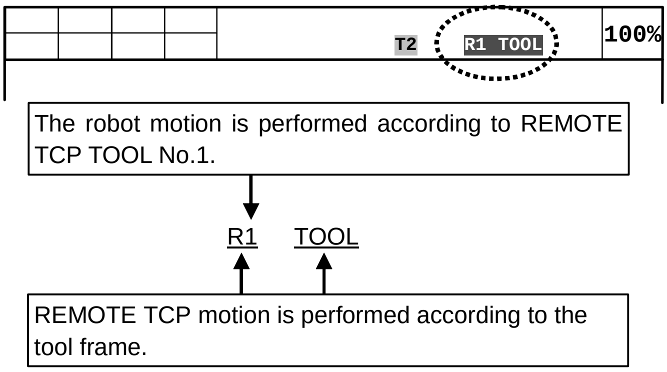
3 Jog the robot.

4 Return from REMOTE TCP jogging mode.

< Change to REMOTE TCP jogging mode >

Please perform the following procedure to enter to REMOTE TCP jogging mode.

1 Press [FCTN] key.
Select “TOGGLE REMOTE TCP”.
The screen is changed as follows.



The robot motion is performed according to REMOTE TCP TOOL No.1.

REMOTE TCP motion is performed according to the tool frame.

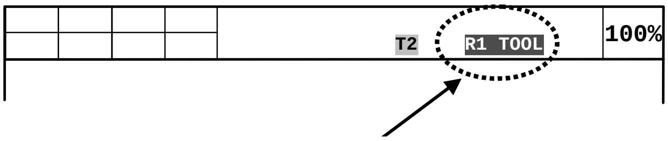
Please perform the following procedure to return from REMOTE TCP jogging mode.

1 Press [FCTN] key.

2 Select “TOGGLE REMOTE TCP”.

Select frame

You change the coordinate system by pressing the COORD key. Please select the frame except for JOINT.



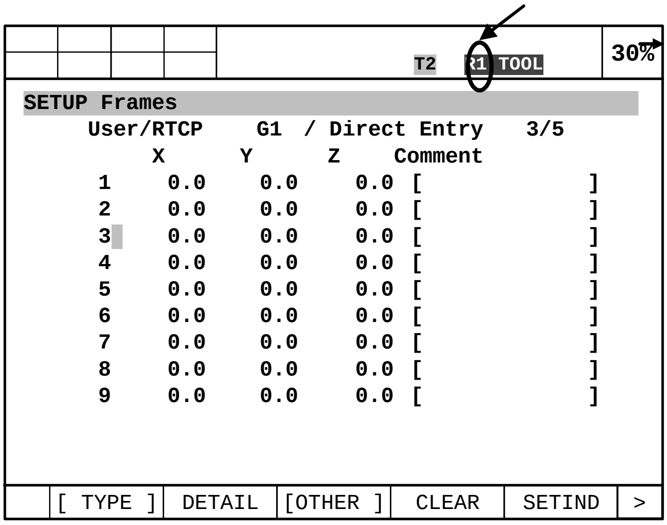
The display status is changed as follows.
JOINT → R1/JFRM → R1/WRLD → R1/TOOL → R1/USER

Select REMOTE TCP frame

When the robot enters to REMOTE TCP mode, you can select REMOTE TCP frame by using how to select the normal frame. And you can change REMOTE TCP frame by the following method, too.

1 Press [FCTN] key.
2 Select “CHANGE RTCP FRAME”.

The number is changed as follows.
1→ 2→ 3→ 4→ 5→ 1



```
SETUP Frames
     User/RTCP     G1  / Direct Entry    3/5
          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 [              ]
      6     0.0    0.0    0.0 [              ]
      7     0.0    0.0    0.0 [              ]
      8     0.0    0.0    0.0 [              ]
      9     0.0    0.0    0.0 [              ]

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

Programming and running by using REMOTE TCP

If you specify REMOTE TCP motion option, the robot moves remote TCP instead of robot TCP.
Please perform the following procedure to specify REMOTE TCP motion option.

1 Move the cursor to end of program line.
2 Press F4, [CHOICE].
3 Select “RTCP”.

Please perform the following procedure to remove REMOTE TCP motion option.

1 Move the cursor to end of program line.
2 Press F4, [CHOICE].
3 Select “No Option”.

You can not use RTCP with JOINT motion.

<Programming example>

```
L    P[1] 100mm/sec FINE RTCP
```

The work is moved to P[1] by the relative speed 100 mm/sec between the work and the remote tool.

```
C    P[1]
     P[2] 100mm/sec FINE RTCP
```

The work is moved to P[2] via P[1] by the relative speed 100 mm/sec between the work and the remote tool.

8 HIGH SENSITIVITY COLLISION DETECTION

High-sensitivity collision detection provides a highly sensitive method to detect that the robot has collided with an object and then stops the robot immediately.
Detection sensitivity of this function is greatly improved than normal collision detection function and this helps to minimize the potential for damage to the end-of-arm tooling and robot.

8.1 SPECIFICATION

1) When collision is detected, the function issues an alarm and stops the robot quickly by decelerating it in such a way that shocks to the robot can be decreased.
2) Since the detection sensitivity has previously been adjusted for each robot type, you can use this function without adjusting it.
3) You can change the detection sensitivity by the program instruction or in the setting screen.
4) You can temporarily enable/disable this function by the program instructions. (It is impossible to disable this function regularly.)
5) You can output signal to a DO when collision is detected.
6) You can output status of the collision detection (enabled / disabled) to a DO.
7) The detection sensitivity is automatically increased during teaching operation.
8) This function is enabled since the power of the robot controller is on.

8.2 SETTINGS

Payload information (Basic)

Set the function with load information and the information about devices installed on the robot.
Since the function uses the load information and device information to detect a collision, it is necessary to set the function with these pieces of information. Be sure to specify the weight of the load, the center of gravity, and the weight of each device on the robot accurately.
If the inertia (shape) of the load is large, it may be necessary to specify the inertia around the gravity center of the load. (If the tool is big, and simply specifying its weight and gravity center does not assure accurate detection, specify its inertia.)
See Section 3.17, “LOAD SETTING” in FANUC Robot series R-30*i*B/R-30*i*B Mate CONTROLLER OPERATOR’S MANUAL (Basic Operation) (B-83284EN) or Section 9, “ LOAD ESTIMATION” in this manual, for how to specify load information.

Collision Guard Setup screen (Optional)

The detection sensitivity has previously been adjusted for each robot type. If you want to change the sensitivity, set in this screen.
Also, If you want to output the status of the collision detection (enabled / disabled), set in this screen.
See Section 8.4, "Collision Guard Setup Screen".

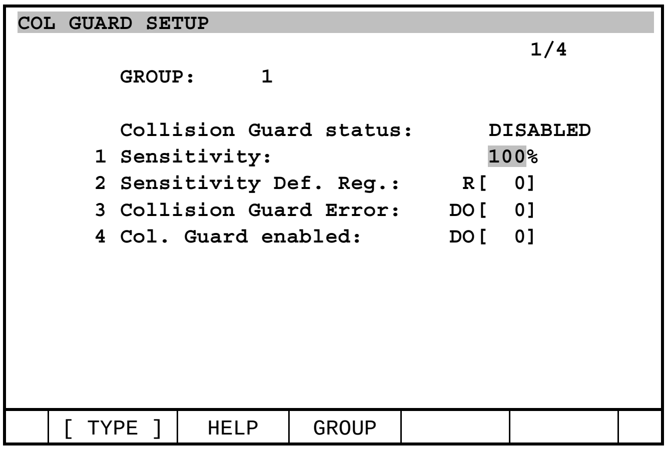
Program instructions (Optional)

If it is previously anticipated that a strong force will be exerted during an operation and users want to temporarily disable collision detection or change the detection sensitivity, insert program instructions.
See Section 8.5, "Program Instructions".

8.3 COLLISION GUARD SETUP SCREEN

Set as below.

1 Press [MENU] key.
2 Select "6 SETUP".
3 Press F1, [TYPE].
4 Select "Coll Guard". The following screen will be displayed.



```
COL GUARD SETUP
                                         1/4
        GROUP:     1

        Collision Guard status:      DISABLED
      1 Sensitivity:                 100%
      2 Sensitivity Def. Reg.:     R[   0]
      3 Collision Guard Error:    DO[   0]
      4 Col. Guard enabled:       DO[   0]
```

Fig. 8.3 Collision guard setup screen

5 Setup each item of Collision Guard Setup screen referring to Table 8.3.

Table 8.3 Setup items of collision guard setup screen

Item	default	Description
Group	1	This item indicates the currently selected group to which the other items in the menu apply. Note that you cannot edit this field directly. Instead, the current group is selected by pressing F3, GROUP and entering the group number you wish to display. You will only be able to select a group that supports High-Sensitivity Collision Detection.
Collision Guard Status	ENABLED	This item specifies whether collision detection is enabled or disabled for the currently selected group. This item cannot be changed in this setup screen.
Sensitivity	100%	This item allows you to set the level of sensitivity for Collision Detection for the currently selected group. Minimum is 1%. Maximum is 200%. The lower the value, the lower the sensitivity. The higher the value, the higher the sensitivity. In some cases, you can decrease the sensitivity value to eliminate false alarms. In some cases, you can increase the sensitivity value to provide faster response.
Sensitivity Def. Reg.	0	This item allows you to specify the register that can be used with the COL GUARD ADJUST teach pendant instruction to adjust the sensitivity of Collision Detection within a program. A register number of 0 indicates that the register is not used.
Collision Guard Error	0	This item allows you to specify a Digital Output that will be turned ON when a Collision Detect Alarm occurs. A value of 0 indicates that no output will be turned on.

Item	default	Description
Col. Guard enabled	0	This item allows you to specify a Digital Output that will be turned on whenever Collision Detection is enabled via this setup menu, and turned off whenever Collision Detection is disabled via this setup menu. A value of 0 indicates that no output will be used.

⚠ **CAUTION**

You can make the collision detection sensitivity insensitive by setting "Sensitivity" less than 100, but in this case, this function cannot work effectively. Please use this function by not decreasing "Sensitivity" as much as possible, and if you decrease "Sensitivity" by necessity, please be very careful.
If robot repeatedly absorbs force as much as collision is detected, the mechanical parts become strained and the life of the robot is shortened. It is desirable to change program to lessen the force instead of changing the sensitivity.

NOTE

It is impossible to set the "Collision Guard Setup" at the group which includes a robot that is not supported "High-Sensitivity Collision Detection" function.

8.4 PROGRAM INSTRUCTIONS

The program instructions below are reserved.

- COL DETECT ON
- COL DETECT OFF
- COL GUARD ADJUST

8.4.1 COL DETECT ON / COL DETECT OFF

These instructions allow you to enable/disable collision detection during program execution.
By default, collision detection is enabled.

- To disable collision detection temporarily, include the "COL DETECT OFF" instruction in a teach pendant program.
- To enable collision detection again, include the "COL DETECT ON" instruction in a teach pendant program.

When the program is finished or aborted, collision detection turns enabled automatically.

Example

```
10: J P[1] 100% FINE
11:  COL DETECT OFF
12: L P[2] 2000mm/sec CNT100
13: L P[3] 2000mm/sec CNT100
14: L P[4] 2000mm/sec CNT100
15:  COL DETECT ON
16: J P[2] 50% FINE
```

This program disables collision detection with lines 12 to 14.

⚠ **CAUTION**

While collision detection is disabled, not only "High-Sensitivity Collision Detection" function but also the basic collision detection function is disabled. Please be careful adequately. For safety's sake, please set the collision detection disable region as short as possible.

NOTE

The COL DETECT ON and COL DETECT OFF instructions will only apply to the motion groups that are included in the group mask of the teach pendant program that calls them. For example, if COL DETECT OFF is used in a teach pendant program that contains group 2 only in its group mask, then Collision Detection will only be disabled for group 2; it will remain enabled for all other groups.

8.4.2 COL GUARD ADJUST

This instruction allows you to adjust the sensitivity of collision detection within a teach pendant program. The sensitivity value set by this instruction will temporarily override what has been set in the COL GUARD SETUP menu.

This instruction can be used in three ways. (See Table 8.4.2.)

Table 8.4.2 COL GUARD ADJUST instruction

number of arguments	the new sensitivity value
0	value of register whose register number is specified in the Collision Guard Setup menu
1 (directly set)	value entered explicitly
1 (indirectly set)	value of set register

⚠ **CAUTION**

You can make the collision detection sensitivity insensitive by setting "Sensitivity" less than 100, but in this case, this function cannot work effectively. Please use this function by not decreasing "Sensitivity" as much as possible, and if you decrease "Sensitivity" by necessity, please be very careful.

If robot repeatedly absorbs force as much as collision is detected, the mechanical parts become strained and the life of the robot is shortened. It is desirable to change program to lessen the force instead of changing the sensitivity.

NOTE

"COL GUARD ADJUST" instruction will apply only to the motion group(s) that is indicated in the group mask of the program that uses COL GUARD ADJUST. For example, if COL GUARD ADJUST is used in a program whose group mask is group 2 only, then the new sensitivity will apply only to group 2; the sensitivity of all other groups will remain unchanged.

If the group mask of the program that uses COL GUARD ADJUST contains a group that does not support High-Sensitivity Collision Detection, a warning message will be posted as follows:

```
MOTN-404 Group # does not support HSCD
```

This message is just a warning to indicate that no action was taken on that group.

To Use the COL GUARD ADJUST instruction without arguments in a TP program, the following procedure should be used:

- Set the register(s) specified in the COLL GUARD SETUP menu for the appropriate group(s) to the new sensitivity value(s).

  NOTE If the TP program is a multiple group program, the appropriate register for each group needs to be set.
- Insert the COL GUARD ADJUST instruction into the TP program at the desired location.

NOTE If the register number in the COL GUARD SETUP menu is zero for one of the groups being adjusted, then an error will be posted.

```
MOTN-400 No Coll. Guard Reg. Defined
```

If a valid register number is specified in the setup menu, but the data in that register is not an integer between 1 and 200, then the following error will be posted:

```
MOTN-401 Coll. Guard Reg. Data Error
```

See the following screen for an example.

```
10: J P[1] 100% FINE
11: R[11]=80
12:  COL GUARD ADJUST
13: L P[2] 2000mm/sec CNT100
14: L P[3] 2000mm/sec CNT100
15: L P[4] 2000mm/sec CNT100
16: R[11]=100
17:  COL GUARD ADJUST
18: J P[5] 50% FINE
```

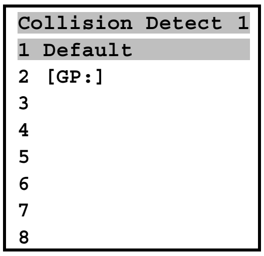
This example would apply to a single group program (say, a group 1 only program). In the example, it is assumed that the register number has been set to 11 for group 1 in the COL GUARD SETUP menu.

Alternatively, the COL GUARD ADJUST instruction can be supplied with the new sensitivity value directly. An example of how this could be done is as follows:

```
10: J P[1] 100% FINE
11:  COL GUARD ADJUST 80
12: L P[2] 2000mm/sec CNT100
13: L P[3] 2000mm/sec CNT100
14: L P[4] 2000mm/sec CNT100
15: R[1]=100
16:  COL GUARD ADJUST R[1]
17: J P[5] 50% FINE
```

To use the COL GUARD ADJUST instruction in this way, after inserting the instruction into the program, move the cursor to the right of the instruction. Then, either type in the new sensitivity value directly, or select F3"INDIRECT" and type in the desired register number.

As stated above, the COL GUARD ADJUST instruction will apply to all groups in the group mask of the TP program that called it. However, if the instruction is used with an argument, the user also has the option to restrict the groups whose sensitivity will be adjusted. This is accomplished by moving the cursor to the argument filed and pressing the F1, GP_MASK key. Doing so will display the following menu at the top of the Teach Pendant screen:



```
Collision Detect 1
1 Default
2 [GP:]
3
4
5
6
7
8
```

Selecting "1" for "Default" means default behavior of the instruction, i.e. all groups in the group mask are adjusted. Selecting "2" inserts a "GP" modifier into the instruction as follows:

COL GUARD ADJUST GP1,2: ##

The cursor will then be on the "##", which is the field where the sensitivity can be directly entered. The user can then move the cursor to the left to highlight the "GP1,2" field. Once this field is highlighted, the function keys will have labels displayed for each available motion group, i.e. F1 will have "GP1", F2 will have "GP2", etc. Pressing the appropriate function key will toggle the appearance of that group number in the "GP1,2" field. Once the "GP1,2" field has been set, only those groups listed in this field will have their sensitivity updated by the instruction.

By default, when a TP program that calls COL GUARD ADJUST is aborted, the sensitivity will be reset to the value specified in the COL GUARD SETUP menu for each group in the program's group mask. This feature can be disabled by setting the following system variable:

```
$HSCDMNGRP[g].$AUTO_RESET = 0
```

where "g" refers to the group number.

8.5 CAUTIONS

1 Collision Detection might detect a false collision when a collision has not occurred in the following cases:
- Payload information has not been set correctly.
- The payload is larger than the maximum payload for the robot, or the inertia of the payload is too large.
- Not enough voltage has been supplied to the controller.
- Low temperature.
- The ACC motion option has been used, causing jerky robot motion.
- Severe motion such as reverse motion which Cnt. is used.
- Linear motion occurs near singularity point where axes revolve in high speed.
  Action: If collision misdetection occurred for the reasons above, try removing these reasons first. If necessarily, insert COL GUARD ON / OFF or COL GUARD ADJUST instructions at the top and bottom of where misdetection occurs for avoiding alarm stop.

2 Collision Detection is disabled in the following cases:
- Soft Float is enabled.
- The robot brakes are on.
- Calibration is not finished.
- While not releasing [SHIFT] key after pressed SHIFT + RESET

3 In order to decrease the force of collision, Collision Detection allows the robot axes to sag away from the collision for a short time after detecting a collision. When this happens, vertical robot axes might fall slightly after detecting a collision, due to the effect of gravity.

9 LOAD ESTIMATION

Load estimation is a function for estimating the weight of the load, such as tool and workpiece, mounted on the hand of the robot.
The function enables the information stated above to be estimated automatically by running the robot.
To use this function, High Sensitive Collision Detection Package option or package option which includes it (ex. Motion Package, Basic Interference Check, Intelligent Interference Check, ARC easy smart quick recovery, Lincoln Asia-Pacific Arc package for Standard EQ, Lincoln Asia-Pacific Arc package) is required. Using the function also requires that your model support the load estimation function. If your model does not support the function, you cannot use it.

This chapter describes load estimation for 6-axis robots. For 5-axis robots, read J5 as J4 and J6 as J5.

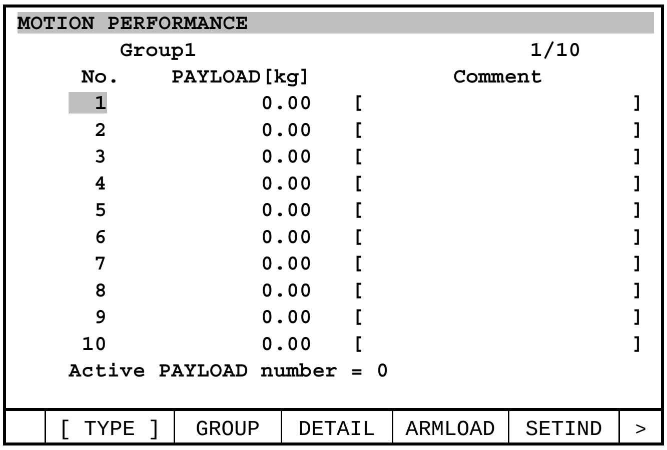
9.1 OPERATING PROCEDURE

Load is estimated in the following flow:
1 Set the range of motion to be subjected to load estimation.
2 Execute load estimation.
Once a mechanical part such as a motor is replaced, it becomes necessary to make calibration.
If no calibration is made after mechanical part replacement, the precision of load estimation becomes lower.

> ⚠ **CAUTION**
> While in execution, load estimation moves J5 and J6 axis with maximum speed (100% joint motion). Do not enter the operating space of the robot. Also, please make sure that all the devices attached to the concerned parts can withstand such motion.

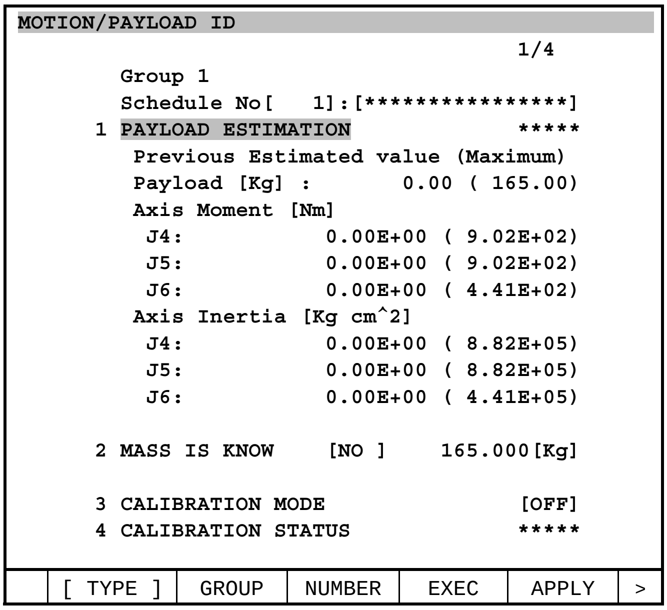
9.2 LOAD ESTIMATION PROCEDURE (for 6-Axis Robots)

This procedure is performed on the load estimation screen.
This screen is accessed from the motion performance screen.
1 Press [MENU] key to display the screen menu.
2 Select “6 SYSTEM” on the next page.
3 Press F1, [TYPE] to display the screen switching menu.
4 Select Motion. The list screen will be displayed. (If any other screen is displayed, press [PREV] key several times until the list screen is displayed.) For a multi-group system, the list screen of another group can be reached by pressing F2, GROUP.



```
MOTION PERFORMANCE
       Group1                              1/10
    No.    PAYLOAD[kg]           Comment
     1            0.00    [                     ]
     2            0.00    [                     ]
     3            0.00    [                     ]
     4            0.00    [                     ]
     5            0.00    [                     ]
     6            0.00    [                     ]
     7            0.00    [                     ]
     8            0.00    [                     ]
     9            0.00    [                     ]
    10            0.00    [                     ]
    Active PAYLOAD number = 0
```

5 Press NEXT, then press F2, IDENT. The load estimation screen will be displayed .



```
MOTION/PAYLOAD ID
                                         1/4
       Group 1
       Schedule No[   1]:[****************]
     1 PAYLOAD ESTIMATION               *****
        Previous Estimated value (Maximum)
        Payload [Kg] :       0.00 ( 165.00)
        Axis Moment [Nm]
         J4:           0.00E+00 ( 9.02E+02)
         J5:           0.00E+00 ( 9.02E+02)
         J6:           0.00E+00 ( 4.41E+02)
        Axis Inertia [Kg cm^2]
         J4:           0.00E+00 ( 8.82E+05)
         J5:           0.00E+00 ( 8.82E+05)
         J6:           0.00E+00 ( 4.41E+05)

     2 MASS IS KNOW    [NO ]    165.000[Kg]

     3 CALIBRATION MODE                 [OFF]
     4 CALIBRATION STATUS               *****
```

6 Place the robot in the position where load estimation is to be performed.

NOTE

1 Only the J5 and J6 axes move during load estimation. The other axes stay in the position where they are when load estimation begins.
The range of motion is defined as an interval between two points specified on estimation position 1 and 2 screens. (See steps 10 and 12.)

2 Put the J5 rotation axis in a horizontal position.
The more vertical posture the J5 rotation axis takes, the lower the precision of estimation becomes.

7 Press F3, NUMBER, and select the load setting schedule No. for which a load estimate is to be set up.

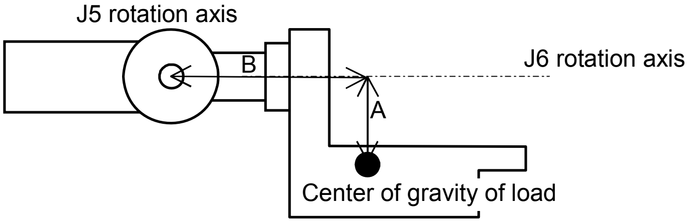
8 If the mass of the load for which load estimation is to be performed is known, move the cursor to line 2, select “YES”, and specify (enter) the mass.

NOTE

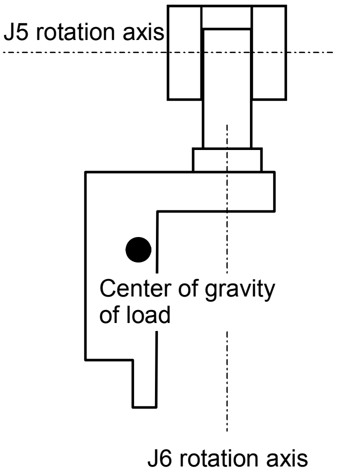
A load with very small mass cannot be estimated. As a rough guide, it requires the load with 5% of maximum load capacity or more.

The estimation precision becomes higher when a mass is specified. Specify the mass as much as possible.
Even if no mass is specified, estimation is possible provided that the following condition is satisfied. However the precision becomes lower.

- The moment around the J5 and J6 axes must be sufficiently high (the mass must be sufficiently great, and the distance A and B, the distance from center of gravity to the rotation axes of J6 and J5, must be sufficiently large.).

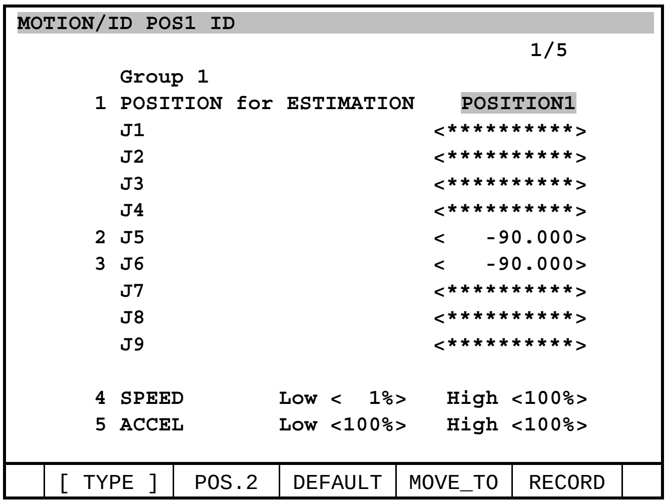


- As for positions set up on estimation position 1 and 2 screens, the center of gravity of the load must be in or near the plane that contains the J5 and J6 rotation axes.



- As for the J6 axis, the interval between points specified on the estimation position 1 and 2 screens must be 180° in terms of angle.

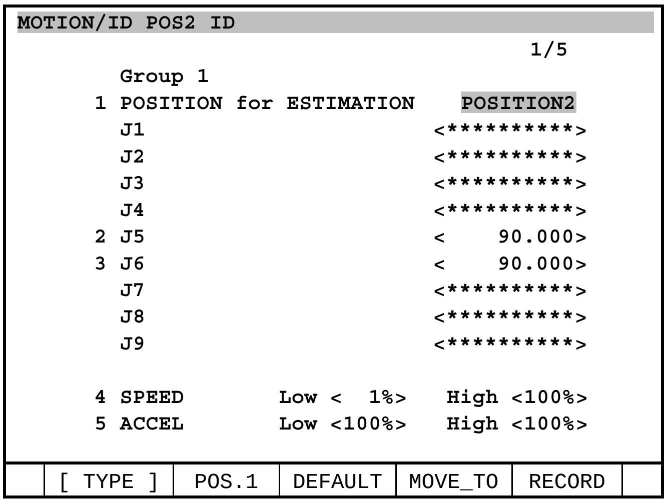
9 Press [NEXT] key, then press F4, DETAIL. The estimation position 1 screen will be displayed.



```
MOTION/ID POS1 ID
                                             1/5
        Group 1
      1 POSITION for ESTIMATION    POSITION1
        J1                        <**********>
        J2                        <**********>
        J3                        <**********>
        J4                        <**********>
      2 J5                        <   -90.000>
      3 J6                        <   -90.000>
        J7                        <**********>
        J8                        <**********>
        J9                        <**********>

      4 SPEED        Low <  1%>    High <100%>
      5 ACCEL        Low <100%>    High <100%>

   [ TYPE ]   POS.2   DEFAULT   MOVE_TO   RECORD
```

10 Specify estimation position 1. (Alternatively, the initial value can be used.)
Specify the positions of the J5 and J6 axes by entering their values directly. Alternatively, move the robot to the desired position by jogging, then press [SHIFT] key + F5, RECORD to record the position.
Now pressing [SHIFT] key + F4, MOVE_TO moves the robot to estimation position 1. Use this procedure to identify the set position.

11 Pressing F2, POS.2 displays the estimation position 2 screen.



```
MOTION/ID POS2 ID
                                             1/5
        Group 1
      1 POSITION for ESTIMATION    POSITION2
        J1                        <**********>
        J2                        <**********>
        J3                        <**********>
        J4                        <**********>
      2 J5                        <    90.000>
      3 J6                        <    90.000>
        J7                        <**********>
        J8                        <**********>
        J9                        <**********>

      4 SPEED        Low <  1%>    High <100%>
      5 ACCEL        Low <100%>    High <100%>

   [ TYPE ]   POS.1   DEFAULT   MOVE_TO   RECORD
```

12 Specify estimation position 2. (Alternatively, the initial value can be used.)
Specify the positions of the J5 and J6 axes by entering their values directly. Alternatively, move the robot to the desired position by jogging, then press [SHIFT] key + F5, RECORD to record the position.
Now pressing [SHIFT] key + F4, MOVE_TO moves the robot to estimation position 2. Use this procedure to identify the set position.

13 Press [PREV] key to return to the estimation screen.

14 Set the teach pendant enable switch to OFF, and press F4, EXEC. The message “Robot moves and estimates. Ready?” will be displayed .

15 Specify whether to execute load estimation. (Selecting “YES” allows the robot to move. Pay sufficient care to avoid danger.)

- To perform load estimation by running the robot, press F4, YES.
- To quit execution, press F5, NO.

16 After low-speed and high-speed operations are finished, load information is estimated. (Operation switches automatically from low speed to high speed. Even when the robot is running at low speed, do not get close to it, because otherwise you may get in a dangerous situation when the robot suddenly starts running at high speed.)

17 Press F5, APPLY to set the estimate at a load setting schedule No. The message “Path and Cycle time will change. Set it?” will be displayed .

18 Specify whether to set the estimate result.

- To set the estimate, press F4, YES.
- Not to set the estimate, press F5, NO.

19 If the value to be set is greater than the maximum allowable load (indicated in parentheses), the message “Load is OVER spec! Accept?” will be displayed. Specify whether to set this value, just as in the step above.

NOTE

If the estimation fails, any or all of the applied results (payload, payload center, payload intertia) would be 0. Please check the PAYLOAD SET screen of the applied schedule number after APPLY.

If mass is very small, mass is not specified, or moment around J5 or J6 is small, estimation might fail. Please refer to NOTE at procedure 8, to see if the load fulfills the conditions necessary to estimate.

9.3 CALIBRATION PROCEDURE (for 6-Axis Robots)

Once a mechanical part such as a motor is replaced, it becomes necessary to make calibration.

If no calibration is made after mechanical part replacement, the precision of load estimation becomes lower. Calibration is controlled, using the load estimation screen. Calibration is started by setting the calibration switch to ON and executing load estimation.

1 Make sure that there is nothing on the hand of the robot.
Calibration must be made without attaching anything to the hand of the robot.

NOTE

If calibration is performed with anything attached to the robot hand, incorrect calibration data is set up, thus hampering a normal estimation.

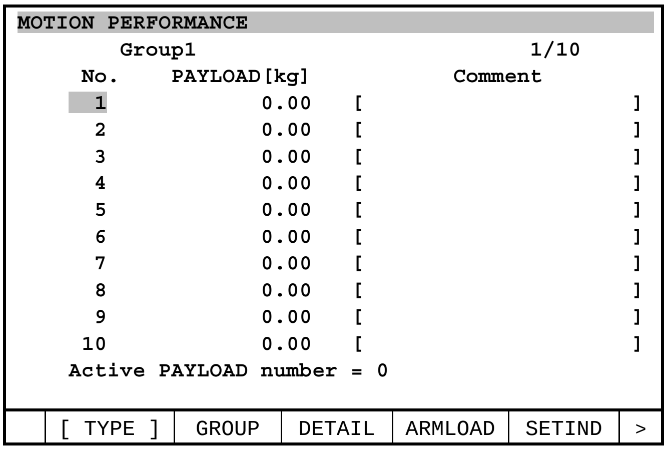
In this case, make calibration again, properly this time.

2 Press [MENU] key to display the screen menu.

3 Select “6 SYSTEM” described on the next page.

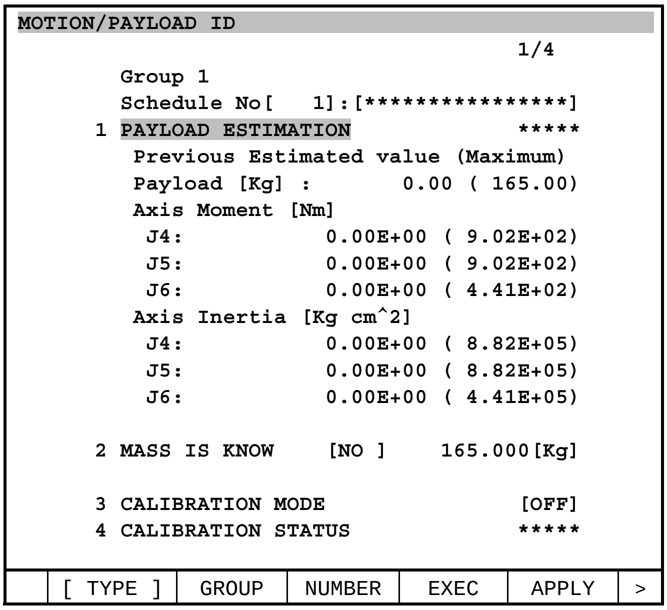
4 Press F1, [TYPE] to display the screen switching menu.

5 Select Motion. The list screen will be displayed. (If any other screen is displayed, press [PREV] key several times until the list screen is displayed.) For a multi-group system, the list screen of another group can be reached by pressing F2, GROUP.



```
MOTION PERFORMANCE
        Group1                          1/10
     No.    PAYLOAD[kg]          Comment
      1           0.00    [                      ]
      2           0.00    [                      ]
      3           0.00    [                      ]
      4           0.00    [                      ]
      5           0.00    [                      ]
      6           0.00    [                      ]
      7           0.00    [                      ]
      8           0.00    [                      ]
      9           0.00    [                      ]
     10           0.00    [                      ]
    Active PAYLOAD number = 0

   [ TYPE ]   GROUP   DETAIL   ARMLOAD   SETIND   >
```

6 Press [NEXT] key, then F2, IDENT. The load estimation screen will be displayed .

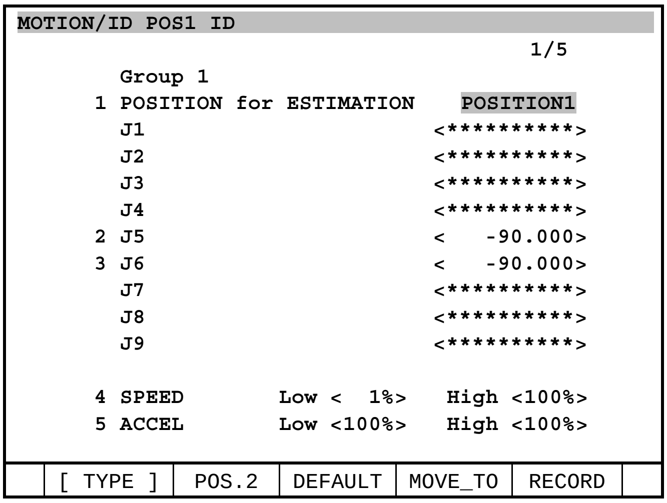


```
MOTION/PAYLOAD ID
                                                1/4
        Group 1
        Schedule No[   1]:[****************]
      1 PAYLOAD ESTIMATION                 *****
         Previous Estimated value (Maximum)
         Payload [Kg] :        0.00 ( 165.00)
         Axis Moment [Nm]
          J4:           0.00E+00 ( 9.02E+02)
          J5:           0.00E+00 ( 9.02E+02)
          J6:           0.00E+00 ( 4.41E+02)
         Axis Inertia [Kg cm^2]
          J4:           0.00E+00 ( 8.82E+05)
          J5:           0.00E+00 ( 8.82E+05)
          J6:           0.00E+00 ( 4.41E+05)

      2 MASS IS KNOW    [NO ]    165.000[Kg]

      3 CALIBRATION MODE                   [OFF]
      4 CALIBRATION STATUS                 *****

   [ TYPE ]   GROUP   NUMBER    EXEC    APPLY   >
```

7 Place the robot in the position where load estimation is to be performed.

> **NOTE**
> 1 Only the J5 and J6 axes move during load estimation. The other axes stay in the position where they are when load estimation begins.
> The range of motion is defined as an interval between two points specified on estimation position 1 and 2 screens. (See steps 9, 10, and 12.)
> 2 Put the J5 rotation axis in a horizontal position.
> The more vertical posture the J5 rotation axis takes, the lower the precision of estimation becomes.

8 Press [NEXT] key, then F4, DETAIL. The estimation position 1 screen will be displayed .



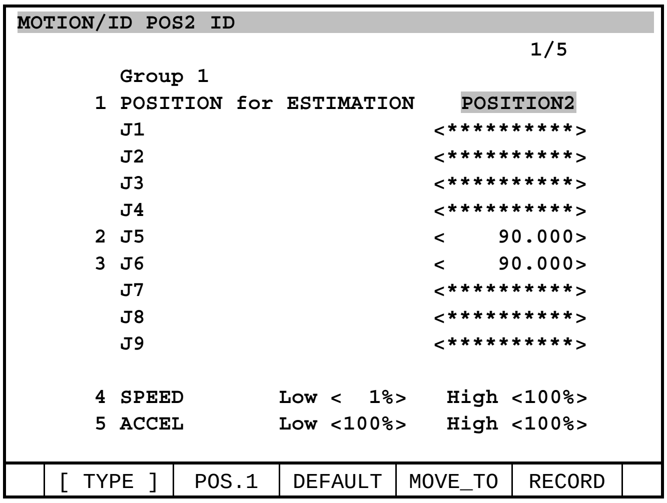
```
MOTION/ID POS1 ID
                                                  1/5
        Group 1
      1 POSITION for ESTIMATION    POSITION1
        J1                        <**********>
        J2                        <**********>
        J3                        <**********>
        J4                        <**********>
      2 J5                        <   -90.000>
      3 J6                        <   -90.000>
        J7                        <**********>
        J8                        <**********>
        J9                        <**********>

      4 SPEED        Low <  1%>   High <100%>
      5 ACCEL        Low <100%>   High <100%>

   [ TYPE ]   POS.2   DEFAULT   MOVE_TO   RECORD
```

9 Specify estimation positions 1 and 2. Try to use default values as much as possible.
Press F3, DEFAULT, and specify default values for estimation positions 1 and 2, speed, and acceleration.

10 Pressing SHIFT + F4, MOVE_TO moves the robot to estimation position 1. Make sure that it is safe to move the robot to estimation position 1.

If it is dangerous to move the robot to estimation position 1, manipulate the J1 to J4 axes by jogging to move the robot to a position where the robot can move safely.

11 Pressing F2, POS.2 displays the estimation position 2 screen.



```
MOTION/ID POS2 ID
                                             1/5
        Group 1
      1 POSITION for ESTIMATION    POSITION2
        J1                       <**********>
        J2                       <**********>
        J3                       <**********>
        J4                       <**********>
      2 J5                       <    90.000>
      3 J6                       <    90.000>
        J7                       <**********>
        J8                       <**********>
        J9                       <**********>

      4 SPEED        Low <  1%>   High <100%>
      5 ACCEL        Low <100%>   High <100%>

   [ TYPE ]   POS.1   DEFAULT   MOVE_TO   RECORD
```

12 Pressing [SHIFT] key + F4, MOVE_TO moves the robot to estimation position 2. Make sure that it is safe to move the robot to estimation position 2.
If it is dangerous to move the robot to estimation position 2, manipulate the J1 to J4 axes by jogging to move the robot to a position where the robot can move safely.
If you moved any of the J1 to J4 axes, press F2, POS.1 to go back to the estimation position 1 screen, and follow this procedure again from step 10.

13 Press [PREV] key to return to the load estimation screen.

14 Move the cursor to CALIBRATION MODE on line 3 to turn it “on”.

NOTE

Once calibration is completed, CALIBRATION MODE becomes “off” automatically.
Do not change CALIBRATION MODE during calibration or load estimation. Otherwise, calibration may be made incorrectly or may not be made at all.

15 Move the cursor to line 4 (so that “EXEC” will be displayed at F4), and set the teach pendant enable switch to OFF, then press “EXEC”. The message “Robot moves and estimates. Ready?” will be displayed .

16 Specify whether to perform load estimation. (Selecting “YES” causes the robot to move. Pay sufficient care to avoid danger.)

- To perform load estimation by running the robot, press F4, YES.
- To quit execution, press F5, NO.

17 After low-speed and high-speed operations are finished, calibration is completed. (Operation switches automatically from low speed to high speed. Even when the robot is running at low speed, do not get close to it, because otherwise you may get in a dangerous situation when the robot suddenly starts running at high speed.)

9.4 OTHER RELATED MATTERS

(1) Motion range

If the motion range between estimation positions 1 and 2 becomes narrower, the estimation precision may get lower.

The actual motion range should preferably be as wide as the default motion range.

(2) Acceleration for motion used in load estimation

The estimation precision is low for the load whose moment inertia is relatively low compared with the maximum allowable load of the robot. This is because the influence by the moment inertia to the torque of the robot motor is weak.

The estimation precision for this light load may be able to be increased by increasing the acceleration used during operation for load estimation.

Try to increase the acceleration by specifying a larger value in “ACCEL - High” on the estimation position 1 and 2 screens; however, do not specify so large a value that vibration becomes serious during operation.

(3) Calibration data

The following system variable holds calibration data.

```
SPLCL_GRP[group].$TRQ_MGN[axis]
    group   :     Group number
    axis    :     Axis number
```

If improper calibration data is set up, for example, by making calibration with a load mounted by mistake, reassigning the previous data to the system variable can restore the previous calibration data.

It is recommended to keep note of the previous calibration data so as to enable restoration.

(4) Payload setting for estimating motion

If payload setting data while in estimating motion is far different from the actual payload, the estimating motion may cause oscillation. To stabilize the estimating motion, it is recommended to set up the payload data by rough calculation before executing estimation.

Setting rough payload data in advance may help enhancing the precision of the estimation, especially for the payload which does not have sufficient moment around the J5 and J6 axes.

10 PAYLOAD OVER AND PAYLOAD SETTING CONFIRM FUNCTION

This function is a part of “Payload identification function” (J669) option.
Although “packages that include Payload identification function” do not include this function.

Payload over and payload setting function adds an elective “PayLoadCheck” in pull-up menu of system screen.
In “PayLoadCheck” screen, you can use payload over confirm function and payload setting confirm function.

Payload Over Confirm Function

This is a function to do not show overload warning when motor is able to move in actual program with the payload that is over specification.
The program to be evaluated has to be executed beforehand. Then this function evaluates the last executed program when F5(OVLD) is selected. And maximum torque of the last executed program and rated torque is compared.

NOTE

Overload warning is disabled when the maximum torque of checked program is lower than rated torque. Note that overload warning is not shown even if maximum torque of the other program is higher than rated torque.

Payload Setting Confirm Function

This function evaluates payload setting accuracy with little motion of robot.
By this function, customers can confirm whether a payload setting is done correctly after a robot is installed on production line.
It may be difficult to move robot widely on production lines, this function roughly estimates a payload by moving J5 and J6 axes slightly and compares with the value of payload setting and shows an error.
When this function is used, robot has to be in the position where J5 and J6 are loaded large moment. Only J5 and J6 have to be loaded almost maximum moment with current payload and other axes have no limitation.
By this function, J5 and J6 axes move between current position+1[degree] and current position–1[degree].

NOTE

Confirm that robot does not interfere if J5 and J6 axes move between current position+1[degree] and current position–1[degree].

Payload setting accuracy will be shown after robot moved.
Payload setting accuracy of payload [1]-[10] is stored.

10.1 LIMITATIONS

Group

This function supports robot at group 1 only (even if multi motion group option is ordered).

Robot Type

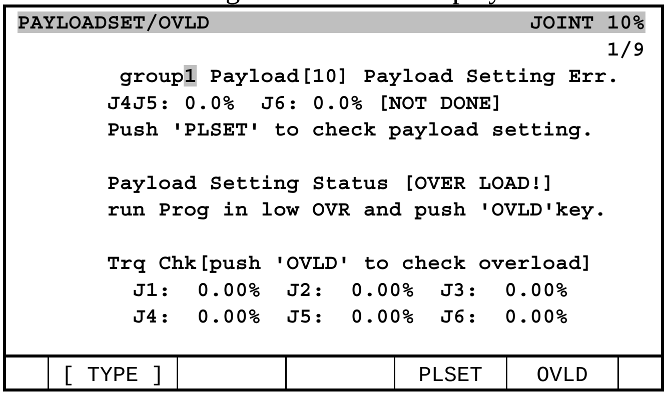
This function supports robot with 6 rotary axes robots. (In supported type, $PLID_GRP.$ROB_TYPE has to be 1 or 2 and $SCR_GRP.$NUM_ROB_AXS has to be 6.)

OK: R-2000*i*B/xxxF, LR Mate 200*i*D etc.
NG: robot with linear axis such as R-2000*i*B/xxxT etc.
NG: robot with 5 axes such as LR Mate200*i*D/7H etc.

10.2 PROCEDURE

10.2.1 Procedure to Show Payload Over and Payload Setting Confirm Screen

1 Press [MENU] key to bring up the screen menu.
2 Select “SYSTEM” on the next page.
3 Press F1“[TYPE]” to display the pull-up menu.
4 Select “PayLoadCheck”. The following screen will be displayed.



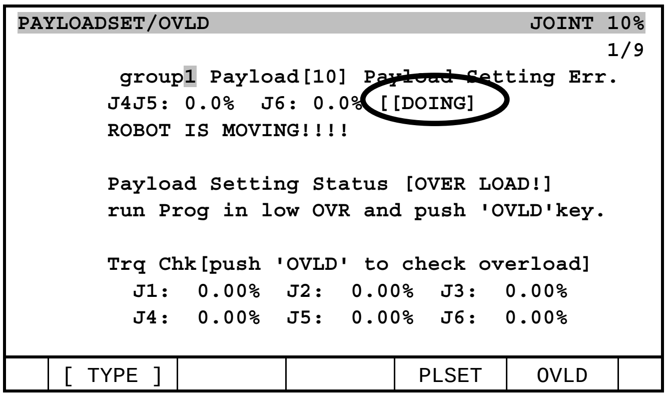
```
PAYLOADSET/OVLD                              JOINT 10%
                                                   1/9
        group1 Payload[10] Payload Setting Err.
       J4J5: 0.0%  J6: 0.0% [NOT DONE]
       Push 'PLSET' to check payload setting.

       Payload Setting Status [OVER LOAD!]
       run Prog in low OVR and push 'OVLD'key.

       Trq Chk[push 'OVLD' to check overload]
         J1:  0.00%  J2:  0.00%  J3:  0.00%
         J4:  0.00%  J5:  0.00%  J6:  0.00%
```

10.2.2 Procedure to Use Payload Setting Confirm (When Payload is Changed)

1 Abort all programs which are running or paused.
2 Move robot to the position where J5 and J6 are loaded almost maximum moment with current payload.

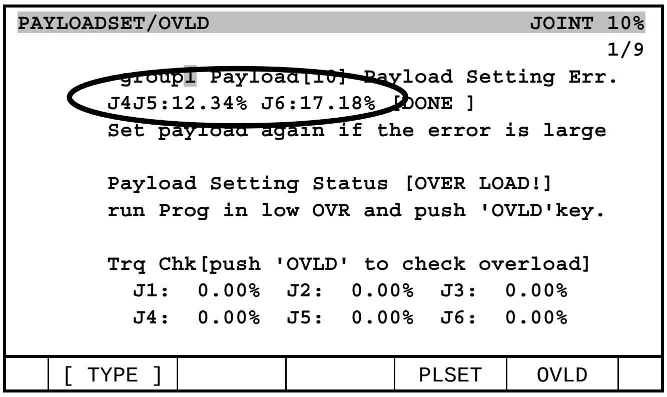
> **NOTE**
> Confirm that robot does not interfere if J5 and J6 axes move between current position+1[degree] and current position–1[degree].

3 Turn off the teach pendant enable switch.
4 Display payload over and payload setting confirm screen. (Refer to the previous section.)
5 Press F4(PLSET) key.
6 Press F3(yes) key.
7 Wait until the right column of the line below “Payload Setting Err.” becomes ”DONE” from “DOING”.(about 30 seconds)



```
PAYLOADSET/OVLD                           JOINT 10%
                                                1/9
        group1 Payload[10] Payload Setting Err.
       J4J5: 0.0%  J6: 0.0% [[DOING]
       ROBOT IS MOVING!!!!

       Payload Setting Status [OVER LOAD!]
       run Prog in low OVR and push 'OVLD'key.

       Trq Chk[push 'OVLD' to check overload]
         J1:  0.00%  J2:  0.00%  J3:  0.00%
         J4:  0.00%  J5:  0.00%  J6:  0.00%
```

8 From “Payload Setting Err.”, you can confirm whether a payload setting is done correctly. Threshold to decide that a payload setting is done correctly is 10%.



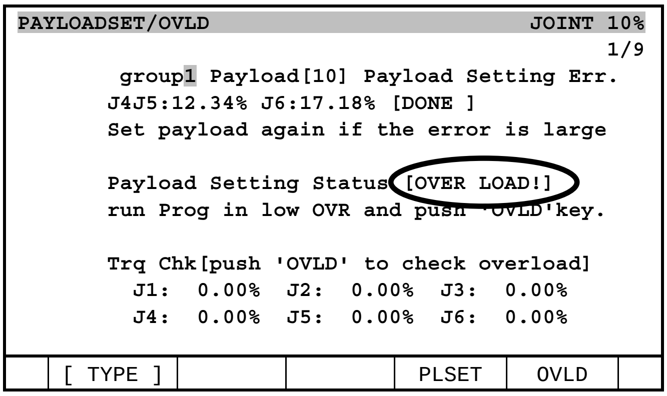
```
PAYLOADSET/OVLD                           JOINT 10%
                                                1/9
        group1 Payload[10] Payload Setting Err.
       J4J5:12.34% J6:17.18% [DONE ]
       Set payload again if the error is large

       Payload Setting Status [OVER LOAD!]
       run Prog in low OVR and push 'OVLD'key.

       Trq Chk[push 'OVLD' to check overload]
         J1:  0.00%  J2:  0.00%  J3:  0.00%
         J4:  0.00%  J5:  0.00%  J6:  0.00%
```

NOTE

You cannot execute payload setting confirm when other program is running or paused.

10.2.3 Procedure to Use Payload Over Confirm (When The Program or Payload is Changed)

1 Please execute torque check by following procedure when “Payload Setting Status” is “OVER LOAD!”.



```
PAYLOADSET/OVLD                           JOINT 10%
                                                1/9
        group1 Payload[10] Payload Setting Err.
       J4J5:12.34% J6:17.18% [DONE ]
       Set payload again if the error is large

       Payload Setting Status [OVER LOAD!]
       run Prog in low OVR and push 'OVLD'key.

       Trq Chk[push 'OVLD' to check overload]
         J1:  0.00%  J2:  0.00%  J3:  0.00%
         J4:  0.00%  J5:  0.00%  J6:  0.00%
```

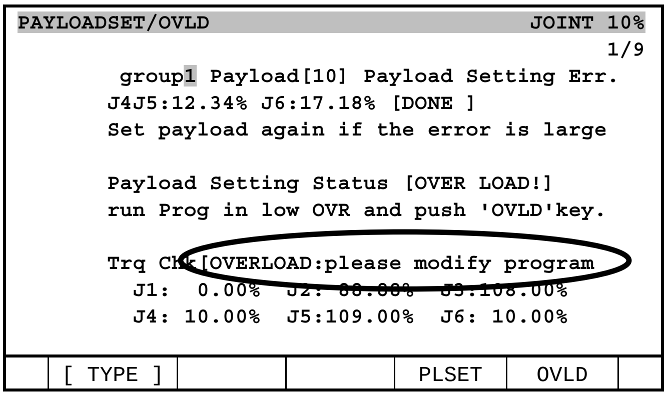
2 The program to be payload over confirmed has to be executed beforehand. Please elect the program to be payload over confirmed. (Push [select] key, select program and press [ENTER] key.)

3 Set override 10% or lower. And execute the program.

4 Then display payload over and payload setting confirm screen. (Refer to the two sections previous.)

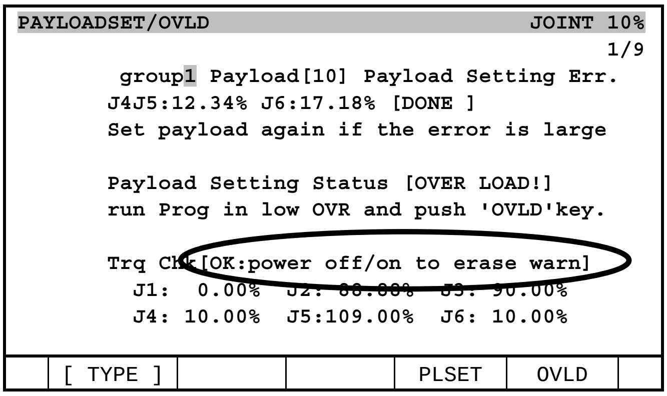
5 Push F5(OVLD) key.

6 Message to ask you to modify program is shown when “Trq Chk” became “OVERLOAD”.



```
PAYLOADSET/OVLD                         JOINT 10%
                                              1/9
        group1 Payload[10] Payload Setting Err.
       J4J5:12.34% J6:17.18% [DONE ]
       Set payload again if the error is large

       Payload Setting Status [OVER LOAD!]
       run Prog in low OVR and push 'OVLD'key.

       Trq Chk[OVERLOAD:please modify program
         J1:  0.00%  J2: 88.00%  J3:108.00%
         J4: 10.00%  J5:109.00%  J6: 10.00%
```

7 Message to ask you to power off and on to disable over load warning is shown when “Trq Chk” became “OK”.



```
PAYLOADSET/OVLD                         JOINT 10%
                                              1/9
        group1 Payload[10] Payload Setting Err.
       J4J5:12.34% J6:17.18% [DONE ]
       Set payload again if the error is large

       Payload Setting Status [OVER LOAD!]
       run Prog in low OVR and push 'OVLD'key.

       Trq Chk[OK:power off/on to erase warn]
         J1:  0.00%  J2: 88.00%  J3: 90.00%
         J4: 10.00%  J5:109.00%  J6: 10.00%
```

NOTE

Overload warning is disabled when the maximum torque of checked program is lower than rated torque. Note that overload warning is not shown even if maximum torques of the other programs are higher than rated torque.

You must confirm that the maximum torque is lower than rated torque in all programs you use. Please change setting of MOTN-171 in setup/accuracy table screen from “NODISP” to “DEFAULT” if you want to enable overload warning again.

Result shown in the screen that is displayed when the program running is not correct. Please display screen again after the program to evaluate has ended.

11 GRAVITY COMPENSATION

Gravity compensation calculates the bending of the robot arm caused by the tool/work on the flange, the equipment on the arm, and the self weight of the arm. Then it compensates the motor position depending on the calculation of the bending, and it improves the absolute position accuracy.
To use this function, Gravity Compensation option (A05B-2600-J649) is required.
This function can not be used with Soft float (A05B-2600-J612).

11.1 SYSTEM VARIABLES

Gravity Compensation

$PARAM_GROUP[group].$SV_DMY_LNK[8]

[Name] Gravity Compensation Enable/Disable
[Meaning] TRUE Gravity Compensation Enable
FALSE Gravity Compensation Disable

Except some specific system configuration, Gravity Compensation is disabled when the robot is shipped. To enable Gravity Compensation, set this variable to TRUE and cycle power.

To set back to be disabled, set this variable to FALSE and cycle power.

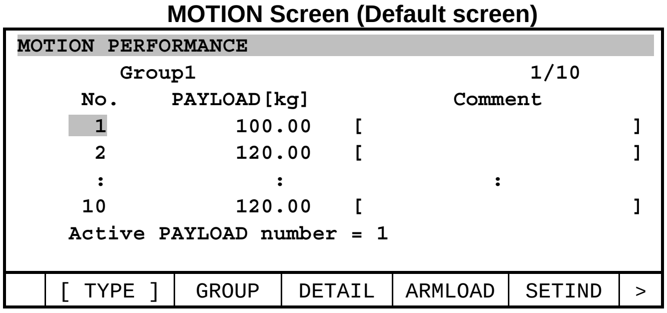
$PARAM_GROUP[group].$MOUNT_ANGLE

[Name] Mount Angle of Robot (Unit: deg)
[Meaning] Set 0 deg for floor mount type, 180 deg for upside down type, or the mount angle for wall mount or angle mount type. Cycle power after setting.

11.2 MOTION SCREEN

1 Payload and armload (equipment on the arm) parameters are set in this screen.
2 This setting screen has three sub-screens. (MOTION screen / PAYLOAD SET screen / ARMLOAD SET screen)
3 This screen is sub-screen in SYSTEM.

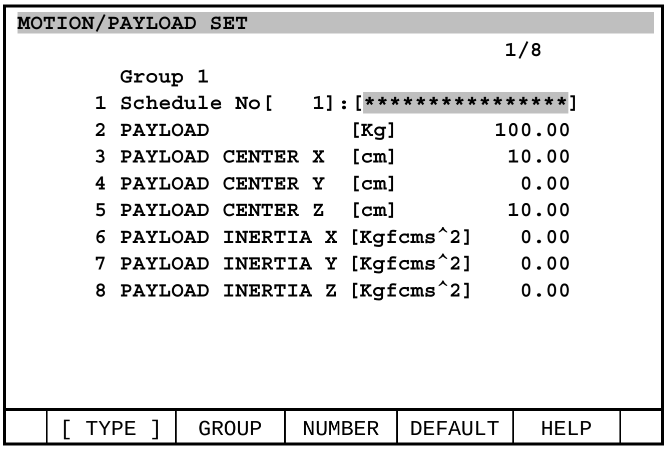
MOTION Screen (Default screen)



```
MOTION PERFORMANCE
        Group1                      1/10
     No.    PAYLOAD[kg]          Comment
       1          100.00   [                     ]
       2          120.00   [                     ]
       :             :               :
      10          120.00   [                     ]
    Active PAYLOAD number = 1
```

4 Payload information (Schedule No.1 to 10) can be setup. Move cursor to the line of one of the schedule numbers, and press F3, DETAIL to enter the payload set screen.

PAYLOAD SET Screen



```
MOTION/PAYLOAD SET
                                         1/8
        Group 1
      1 Schedule No[   1]:[****************]
      2 PAYLOAD            [Kg]       100.00
      3 PAYLOAD CENTER X   [cm]        10.00
      4 PAYLOAD CENTER Y   [cm]         0.00
      5 PAYLOAD CENTER Z   [cm]        10.00
      6 PAYLOAD INERTIA X [Kgfcms^2]    0.00
      7 PAYLOAD INERTIA Y [Kgfcms^2]    0.00
      8 PAYLOAD INERTIA Z [Kgfcms^2]    0.00
```

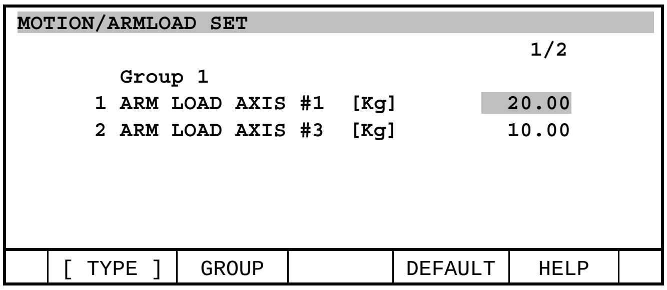
5 Setup the payload, payload center, and payload inertia. X, Y, and Z directions in this screen mean X, Y, and Z axes of the default (the settings are all 0) tool frame.
After the value is input, the message “Path and Cycle time will change. Set it ?” is displayed. Please input F4, YES or F5, NO.

6 To enter the payload set screen of the other schedule number, press F3, NUMBER. To enter the screen for other group, press F2, GROUP. (Only in the multi-group system)

7 Press [PREV] key to go back to the motion screen (default screen). Press F5, SETIND and input the schedule number to use.

8 Press F4, ARMLOAD in the motion screen (default screen) to enter the armload set screen.

ARMLOAD SET Screen



```
MOTION/ARMLOAD SET
                                           1/2
        Group 1
      1 ARM LOAD AXIS #1   [Kg]        20.00
      2 ARM LOAD AXIS #3   [Kg]        10.00
```

9 Setup the armload on axis #1 and axis #3.
After the value is input, the message “Path and Cycle time will change. Set it ?” is displayed. Please input F4, YES or F5, NO.
After setting up the armload, cycle power.

11.3 MASTERING

⚠ **CAUTION**
From R-30*i*B Controller, we can execute mastering even when Gravity Compensation is ENABLED. Because the procedure has been changed from that of R-30*i*A Controller, please read and understand this section before mastering execution.

11.3.1 “Normal Mastering” and “GC Mastering”

You have 2 types of mastering method.

(1) Normal Mastering : Mastering with Gravity Compensation DISABLED

(2) GC Mastering : Mastering with Gravity Compensation ENABLED

- Basically you should choose the same method as the one which you have chosen at the previous mastering. This is because the result of (1) and (2) is different from each other.
- Gravity Compensation works with either mastering result. But you can achieve better absolute position accuracy with GC Mastering.
- If you switch the method from Normal Mastering to GC Mastering, TCP will shift a little. Therefore, please confirm in advance that TCP shift will cause no problem. When you do mastering for switching, use Vision Mastering or Jig Mastering (in other words, don’t use Single Axis Mastering). For details of Vision Mastering, refer to “FANUC Robot series R-30*i*B/R-30*i*B Mate CONTROLLER *i*RCalibration OPERATOR’S MANUAL” (B-83724EN).

Mastering method at the shipment of the robot

For some specific system configuration, GC Mastering has been done at the shipment of the robot. You can find this information in the “Inspection Data Sheet” which is attached to your robot controller.

“Mastering with Gravity Compensation” in Inspection Data Sheet	Mastering method at the shipment
“Yes”	GC Mastering
“No”	Normal Mastering

11.3.2 How to Choose Mastering Method

The following system variable indicates which mastering method the current mastering counts are derived from.

$DMR_GRP[group].$GRAV_MAST

Value	Description	Mastering method you should choose
1	Current mastering data is derived from GC Mastering.	• Choose GC Mastering.
0	Current mastering data is derived from Normal Mastering.	• Basically, choose the Normal Mastering. • If you need to improve absolute position accuracy, choose GC Mastering.
-1	Unknown (not set).	• Choose the same one as “Mastering method at the shipment of the robot” which is described above.

This system variable is updated every time you execute mastering.

11.3.3 Mastering Procedure

Mastering procedure for Normal Mastering

1 If Gravity Compensation is ENABLED, disable it.
2 Perform mastering.
3 If you have disabled Gravity Compensation in step 1, enable it again.

Mastering procedure for GC Mastering

1 Perform mastering with Gravity Compensation ENABLED.
* Please refer to the next subsection for guidance.

After mastering execution, record (write down) the value of $DMR_GRP[group].$GRAV_MAST together with mastering counts ($DMR_GRP[group].$MASTER_COUN[ ]).

NOTE

.When you use Gravity Compensation function, you must treat $DMR_GRP[group].$GRAV_MAST as a part of mastering data. So please remember to record (write down) the value of this system variable together with the mastering counts. You need this value if we manually enter the mastering data to system variable.

```
Example of mastering data
$DMR_GRP[1].$MASTER_COUN[1] = 137859
                        .$MASTER_COUN[2] = 22309853
                        .$MASTER_COUN[3] = 30596
                        .$MASTER_COUN[4] = 810348
                        .$MASTER_COUN[5] = 87291
                        .$MASTER_COUN[6] = 23124
$DMR_GRP[1].$GRAV_MAST = 1
```

In addition to $MASTER_COUN[ ], $GRAV_MAST is also a part of mastering data.

11.3.4 Guidance of GC Mastering

Keep the following rules of GC Mastering.

1 Before you perform GC Mastering, set correct payload parameters via MOTION screen.

2 When you write down joint angles of a reference position which will be used for mastering, do it under the following condition;
- Gravity Compensation is ENABLED, and
- Calibration is completed, and
- Correct payload parameters are set.

3 When you perform Single Axis Master, keep the following rules;

- Just to be safe, write down the Mastering Data ($DMR_GRP[group].$MASTER_COUN[axis]) of all axes before you perform mastering.
- If there are more than one axes which need to be mastered, you must select and master them all at once.

  ex) Single Axis Master after J2, J4 and J5 motor replacement
  You must master J2, J4 and J5 all at once.
  You cannot do master solo J2 when J4 and J5 remain to be mastered.
  * If you have no way to master multiple axes all at once, work around in the following way;
  (NOTE: Master in ascending order. In other words, from robot base to flange)
  (1) First, master J2, J4 and J5 all at once. You need to master J2 correctly. At this time, you don’t need to master J4 and J5 correctly, but master them moderately correctly. In other words, do temporary master for J4 and J5.
  (2) Next, master J4 correctly.
  (3) Finally, master J5 correctly.
- If you move no-need-master axes to the angle which you have wrote down for a reference position in advance, do it after performing temporary mastering of need-master axes.

  ex) Single Axis Master using reference angle of J1,J2,J4,J5,J6 after J3 motor replacement
  (1) First, master J3 moderately correctly, then do CALIBRATE.
  (2) Next, move no-need-master axes (J1,J2,J4,J5,J6) to the angle which you have wrote down for a reference position.
  (3) Finally, master J3 correctly.
- If you perform “mastering after temporary mastering” as described above, master count of the axis which has an interaction with the target axis of mastering will change. This causes unexpected position change. Therefore, after performing mastering, verify that master count of no-need-master axis keeps the original value. If this value has been changed, enter the original value which you have written down in advance.

12 OPERATION LOG BOOK

Operation Log Book automatically records teach pendant operations and alarms in a buffer that can be displayed in the Log Book menu of the teach pendant or saved as a text file. Some operations can include a screen image to help in analyzing the event.

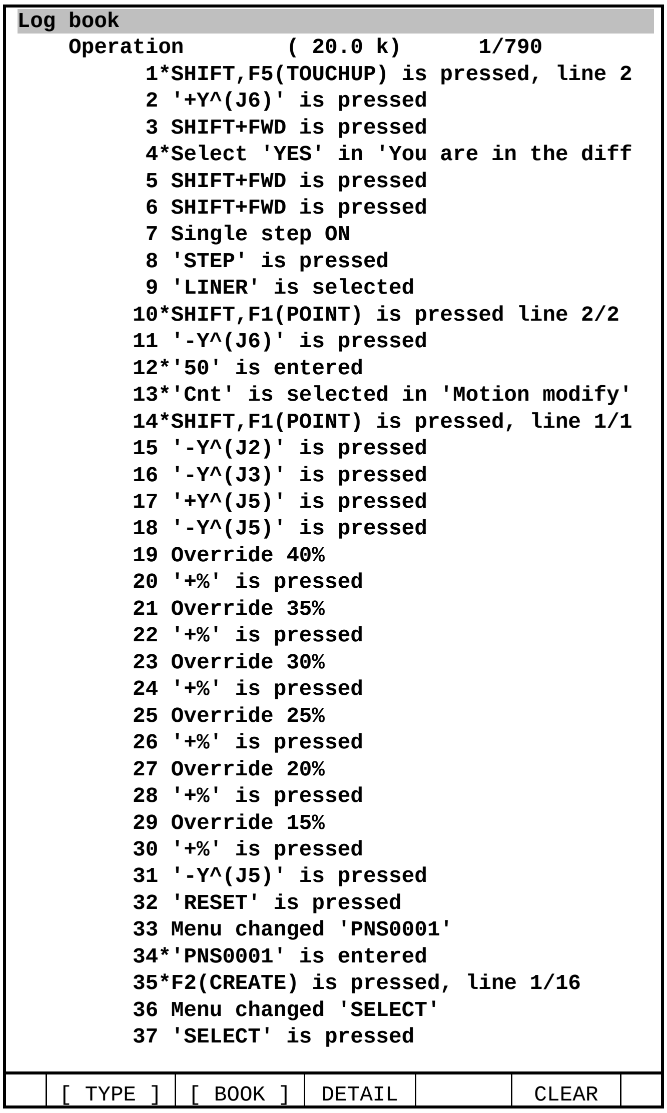
A system can have multiple Log Books and you can specify which events are recorded in each Log Book. In this way, frequent and infrequent events can be recorded to separate Log Books.

When a Log Book buffer is full and a new record is added, the oldest record is deleted. The number of events that can be stored in a Log Book buffer varies based on the type of events saved because not all events are the same size. You can change the size of the buffer by setting a system variable.

To use function, Operation Log Book option (J695) is required.

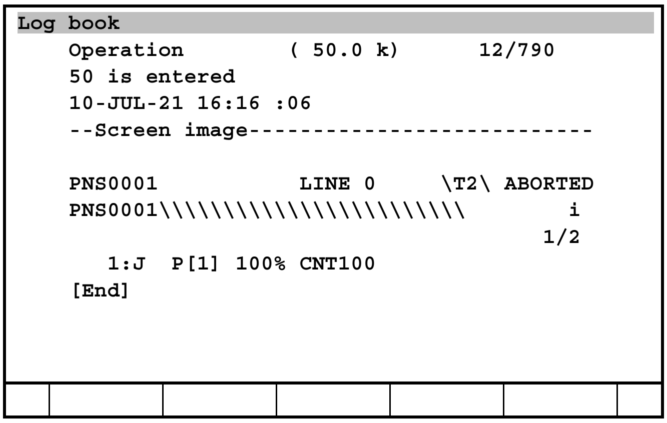
Note : This function is independent of the password log function.

See the following example of a Log Book.



```
Log book
    Operation        ( 20.0 k)      1/790
          1*SHIFT,F5(TOUCHUP) is pressed, line 2
          2 '+Y^(J6)' is pressed
          3 SHIFT+FWD is pressed
          4*Select 'YES' in 'You are in the diff
          5 SHIFT+FWD is pressed
          6 SHIFT+FWD is pressed
          7 Single step ON
          8 'STEP' is pressed
          9 'LINER' is selected
         10*SHIFT,F1(POINT) is pressed line 2/2
         11 '-Y^(J6)' is pressed
         12*'50' is entered
         13*'Cnt' is selected in 'Motion modify'
         14*SHIFT,F1(POINT) is pressed, line 1/1
         15 '-Y^(J2)' is pressed
         16 '-Y^(J3)' is pressed
         17 '+Y^(J5)' is pressed
         18 '-Y^(J5)' is pressed
         19 Override 40%
         20 '+%' is pressed
         21 Override 35%
         22 '+%' is pressed
         23 Override 30%
         24 '+%' is pressed
         25 Override 25%
         26 '+%' is pressed
         27 Override 20%
         28 '+%' is pressed
         29 Override 15%
         30 '+%' is pressed
         31 '-Y^(J5)' is pressed
         32 'RESET' is pressed
         33 Menu changed 'PNS0001'
         34*'PNS0001' is entered
         35*F2(CREATE) is pressed, line 1/16
         36 Menu changed 'SELECT'
         37 'SELECT' is pressed
```

The lines marked with an "*" have an associated screen image, as seen in the example of line 12 below. The screen image shows the previous value of the changed item.



```
Log book
    Operation         ( 50.0 k)      12/790
    50 is entered
    10-JUL-21 16:16 :06
    --Screen image--------------------------

    PNS0001           LINE 0     \T2\ ABORTED
    PNS0001\\\\\\\\\\\\\\\\\\\\\\\        i
                                        1/2
       1:J  P[1] 100% CNT100
    [End]
```

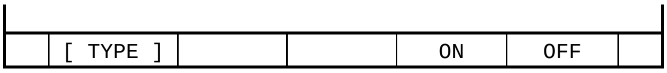
General Limitations

Note the following general limitations for Log Book:

- Log Book does not support certain *i*Pendant operations. See following examples.
  - Selecting a link on a web page
  - Operations in screens for *i*Pendant only. For example, operations in browser, panel setup, *i*Pendant setup screen.
- Operations are not recorded in controlled start just after initial start. Cold start must be performed to begin logging events.
- Operations in the configuration menu are not recorded.
- If you change the size of a log book, all data in the log book are lost.
- Some events are cyclically monitored for logging. Some of the following operations may not be recorded if they occur faster than the monitoring cycle:
  - Override (“Override x%”)
  - Coordination (JOINT coordinate, User Coordinate etc.)
  - Single step (Single step ON/OFF)
  - Motion group (“Motion group x”)
  - Sub group (“Sub group ROBOT/EXT”)
  - User frame number (“User frame x”)
  - Tool frame number (“Tool frame x”)
- If you change the dictionary, some entries in the LOG BOOK screen cannot be read. To read such log, you must use the language that was used when the log was recorded.
- If passwords are enabled and $LOGBOOK.$LOG_ENT is a valid book number when you log in, your password input is recorded as “’x’ is entered”. Passwords for other functions are also recorded. To prevent passwords from being recorded, you can use screen filtering to filter out the screens that contain passwords. Refer to Section 12.4 Operations.
- ‘y’ of “’x’ is selected in ‘y’ window” is based on the 1st line of prompt window. If the 1st line is blank, y contains nothing. Please refer to screen image for analysis.
- If you press a function key that has no label, the key number followed by empty parentheses will be recorded.

  Example

  If the function key line is as follows:

  ```
   [ TYPE ]                ON      OFF
  ```

  F3 has no label. If you press F3 and $LOGBOOK $LOG_FNKEY is a valid value, “F3( )” is pressed” is recorded.
- If you change the remote TCP number in remote TCP jog mode, this is recorded as a change of user frame.
- If you changed current JOG coordination by parameter instruction, the change is recorded when the group is selected.

- Screen image doesn’t support double height font, which is used in the “on the fly” screen of ArcTool. It is recorded as two lines which have the same characters.
  Example
  0.0 Amps
  in “On the fly” screen is recorded as

```
0.0 Amps
0.0 Amps
```

- For KAREL read instructions, screen image is recorded just after input.
- KCL is not supported.
- If the TP and CRT are used at the same time, log book records the operations of both of them. It might be difficult to distinguish between TP and CRT operations in a log.
- Screen changes caused by the automatic backup function are not recorded.
- “’x’ is saved” and “’x’ is loaded” may be recorded by internal process.

12.1 RECORDED EVENTS

The following is the list of all events that can be recorded by the Log Book function.
Recording of individual events can be disabled by setting the system variable specified in the 'Parameter' column. If it is 0, the event is not recorded. Some of them are disabled by default. If the system variable is not 0, the value indicates the Log Book No. to which the event is recorded.
Some events have a screen image. This information helps to analyze the record. However, screen images take a lot of space in the Log Book; you can disable the recording of screen images by setting the system variable specified in the 'Save screen' column.

Refer to “12.3 Operations” later in this section.

Table 12.1 Events recorded by log book

Message	Event	Parameter (Default)	Save Screen (Default)
Alarm history	When an alarm occurs, the alarm message is recorded. This record is the same as the record of the alarm history menu. To choose the alarms to be recorded, the 'filtering' function is provided.	$LOGBOOK. $LOG_ER (1)	None
'x' is entered	When a value or a word is entered, this is recorded. The 'x' in the message is the entered value or word. If screen image is enabled for this event, it will contain the previous value. Values will be recorded even if they are invalid. If you cancel an input, it is not recorded.	$LOGBOOK. $LOG_ENT (1)	$LOGBOOK. $IMG_ENT (TRUE)
'x' is selected 'x' is selected in 'y' menu	When a menu item is selected, this is recorded. The 'x' in the message is the selected item. If the menu has a title, the 'y' in the message shows the title. If screen image is enabled for this input, it will show the screen just before opening the menu.	$LOGBOOK. $LOG_SEL (1)	$LOGBOOK. $IMG_SEL (TRUE)

Message	Event	Parameter (Default)	Save Screen (Default)
'x' is selected in 'y' window	When an item is selected in warning window, this is recorded. The 'x' in the message is the selected item. The 'y' in the message is the beginning of the message in the warning window. If screen image is enabled for this event it will show the warning window.	$LOGBOOK. $LOG_WIN (1)	$LOGBOOK. $IMG_WIN (TRUE)
'x' is selected in MENU 'x' is selected in FCTN	When an item is selected by [MENU] key or FCTN key, this is recorded. The 'x' in the message is the selected item.	$LOGBOOK. $LOG_MENU (1)	None
JOG menu TOOL 'x' JOG menu USER 'x' JOG menu JOG 'x' JOG menu GROUP 'x' JOG menu ROBOT JOG menu EXT	Operations of the JOG menu are recorded with these messages.	$LOGBOOK. $LOG_JGMN (1)	None
Menu changed 'x'	When a menu is changed, this is recorded. The 'x' in the message is the title of the new menu. Changes to sub menus such as Config or Detail are not recorded.	$LOGBOOK. $LOG_MNCHG (1)	None
'x' is pressed	When a key is pressed, this message is recorded with the key name. It is not recorded when the key is released. If SHIFT is held when the key is pressed, the word 'SHIFT,' is added to the key name (except SELECT, EDIT, DATA). The keys are grouped as follows, and you can disable recording of each group.		
	F1, F2, F3, F4, F5	$LOGBOOK. $LOG_FNKEY (1)	$LOGBOOK. $IMG_FNKEY (TRUE)
	+X(J1), -X(J1), +Y(J2), -Y(J2), +Z(J3), -Z(J3), +X^(J4), -X^(J4), +Y^(J5), -Y^(J5), +Z^(J6), -Z^(J6) +(J7), -(J7), +(J8), -(J8)	$LOGBOOK. $LOG_JGKY (1)	None
	SELECT, EDIT, DATA,	$LOGBOOK. $LOG_PRGKEY (1)	None
	UF1, UF2, UF3, UF4, UF5, UF6, UF7	$LOGBOOK. $LOG_UFKY (1)	None
	+%, -%, COORD	$LOGBOOK. $LOG_OVRKY (1)	None
	FWD, BWD	$LOGBOOK. $LOG_FWDKY (1)	None
	HOLD	$LOGBOOK. $LOG_HLDKY (1)	None
	STEP	$LOGBOOK. $LOG_STPKY (1)	None
	PREV	$LOGBOOK. $LOG_PRVKY (1)	None

Message	Event	Parameter (Default)	Save Screen (Default)
'x' is pressed	ENTER	$LOGBOOK. $LOG_ENTKY (1)	None
	ITEM	$LOGBOOK. $LOG_ITMKY (1)	None
	RESET	$LOGBOOK. $LOG_RSTKY (1)	None
Override x%	When override is changed, this is recorded. The 'x' in the message is the new override value. This records all change of override by any method. For example, '+%' key, TP is enabled, override instruction of program.	$LOGBOOK. $LOG_OVR (1)	None
JOINT coordinate USER coordinate TOOL coordinate JOG coordinate PATH coordinate	When coordinate is changed, this is recorded. This records all change of coordinate by any method.	$LOGBOOK. $LOG_CRD (1)	None
Single step ON Single step OFF	When single step is changed, this is recorded. This records all change of single step by any method.	$LOGBOOK. $LOG_STEP (1)	None
Motion group x	When motion group is changed, this is recorded. The 'x' in the message is the new motion group. This records all change of motion group by any method.	$LOGBOOK. $LOG_GRP (1)	None
Sub group ROBOT Sub group EXT	When sub group is changed, this is recorded. This records all change of sub group by any method.	$LOGBOOK. $LOG_SGRP (1)	None
User frame x	When user frame number is changed, this is recorded. The 'x' in the message is the new user frame number. This records all change of user frame number by any method.	$LOGBOOK. $LOG_UF (1)	None
Tool frame x	When tool frame number is changed, this is recorded. The 'x' in the message is the new tool frame number. This records all change of tool frame number by any method.	$LOGBOOK. $LOG_UT (1)	None
Save file x Load file x	File x is saved. File x is loaded. These events are ONLY for file save/load in following screens. • File screen • Program list screen	$LOGBOOK. $LOG_FILE (1)	None
WAIT is released (x, y)	The waiting WAIT instruction is canceled by WAIT RELEASE in program x line y.	$LOGBOOK. $LOG_WTRLS (0)	None
Create program x Delete program x Write line x of y Delete line x of y	TP program x is created. TP program x is deleted. Line x of TP program y is written. Line x of TP program y is deleted.	$LOGBOOK. $LOG_PGCHG (0)	None
Write P[x] of y	Position data P[x] of TP program y is written. These are also recorded when program is changed internally.	$LOGBOOK. $LOG_SETPOS(0)	None

Message	Event	Parameter (Default)	Save Screen (Default)
TP 'x' ON TP 'x' OFF	This records the low level key operation. All key operations of Teach Pendant are recorded. When a key is pressed, 'TP x ON' is recorded. When a key is released, 'TP x OFF' is recorded. ENABLE switch and E-STOP are also recorded. [SHIFT] key is treated as normal key. CRT operations are not recorded by this event. 'x' in the message is key name, the following keys are recorded. ENABLE, ESTOP PREV, F1, F2, F3, F4, F5, NEXT SHIFT, MENU, SELECT, EDIT, DATA,FCTN, *i* UP, DOWN, LEFT, RIGHT, DISP RESET, STEP, BACKSPACE, ITEM, HOLD, FWD, BWD, COORD, +%, -%, GROUP 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -/,, ., ENTER, HELP/DIAG, UF1, UF2, UF3, UF4, UF5, UF6, UF7, +X(J1), -X(J1), +Y(J2), -Y(J2), +Z(J3), -Z(J3), +X^(J4), -X^(J4), +Y^(J5), -Y^(J5), +Z^(J6), -Z^(J6), +(J7), -(J7), +(J8), -(J8)	$LOGBOOK. $LOG_TPKY (0)	None
Cold start Power failure recovery Controlled start	Start mode is recorded.	$LOGBOOK. $LOG_STMD(1)	None

12.2 SETTING UP BOOK

A system can have 16 books at a maximum. User can setup each book by system variable. Followings are items the user can setup.

- Title
- Size
- Memory to store buffer
- Visible/invisible

Table 12.2 System variables to setup book

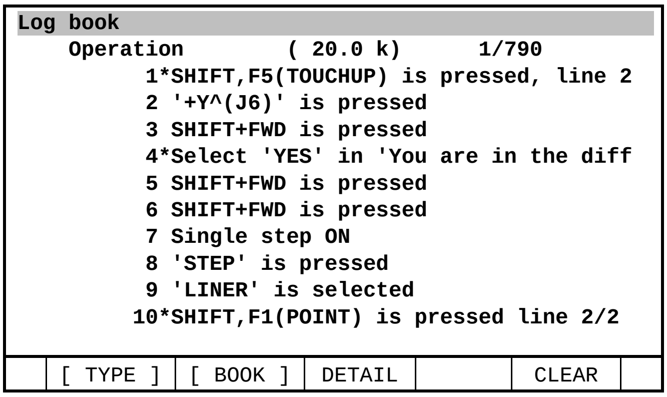
System variable name	Type	Default	Range	Description
$LOG_BUFF[1-16]. $TITLE	String	[1-16] ‘’	Max 13 chars	Title of every buffer. This string is displayed on the top line of LogBook menu. It also be displayed in pop up menu of F2([BOOK]) in LogBook menu. If the value is ‘’, the default title is displayed. Default title of BOOK1 is ‘Operation’. Default title of ‘BOOK2 is ‘I/O’. Default title of BOOK3 is ‘BOOK 3’.

System variable name	Type	Default	Range	Description
$LOG_BUFF[1-16]. $SIZE	Integer	[1] 50 [2] 100 [3-16] 0	0, 2147483647	Size of the buffer to save log. Unit is K Bytes. One record takes about 300 bytes. If this value is too big, maximum available size of Log Book buffer is allocated automatically. If there is not enough memory to make Log book buffer, SYST – 188 WARN “book(%d) was not created” is posted and the book is not created. Cycle power is needed to enable the change of this variable.
$LOG_BUFF[1-16]. $MEM_TYPE	Integer	[1] 0 [2] 1 [3-16] 0	0, 2147483647	Memory type of every buffer. 0: In SRAM, Log is kept at power down. 1: In DRAM, Log is cleared at power down.
$LOG_BUFF[1-16]. $VISIBLE	Boolean	[1] TRUE [2] FALSE [3-16] TRUE	FALSE, TRUE	If FALSE, the buffer is not displayed in pop up menu of F2([BOOK]) in LogBook menu.

12.3 OPERATIONS

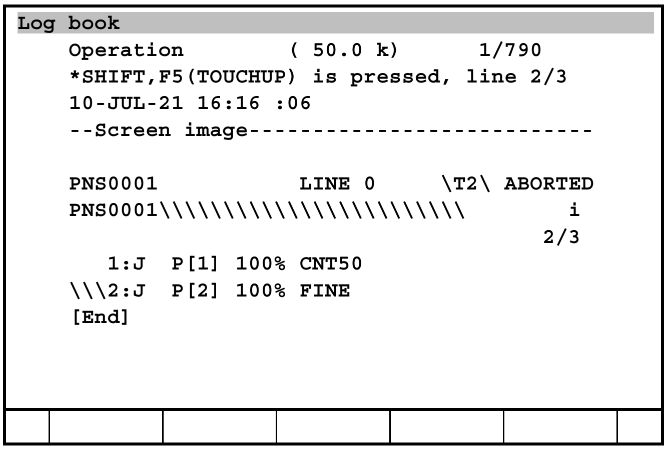
Displaying the Log Book Screen

1 Press [MENU] key.
2 Select 4, ALARM.
3 Press F1, [TYPE].
4 Select Log Book. The following screen will be displayed.



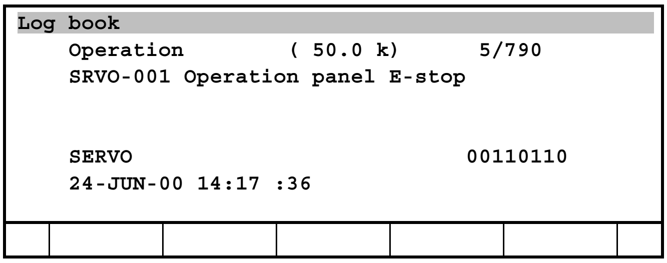
```
Log book
    Operation        ( 20.0 k)      1/790
          1*SHIFT,F5(TOUCHUP) is pressed, line 2
          2 '+Y^(J6)' is pressed
          3 SHIFT+FWD is pressed
          4*Select 'YES' in 'You are in the diff
          5 SHIFT+FWD is pressed
          6 SHIFT+FWD is pressed
          7 Single step ON
          8 'STEP' is pressed
          9 'LINER' is selected
         10*SHIFT,F1(POINT) is pressed line 2/2
```

5 To change to a different Log Book, press F2, BOOK.
6 To view the screen image for a line that is marked with a “*”, select the line and press F3, DETAIL. The following screen will be displayed:



```
Log book
    Operation        ( 50.0 k)      1/790
    *SHIFT,F5(TOUCHUP) is pressed, line 2/3
    10-JUL-21 16:16 :06
    --Screen image---------------------------

    PNS0001           LINE 0     \T2\ ABORTED
    PNS0001\\\\\\\\\\\\\\\\\\\\\\\\        i
                                         2/3
       1:J  P[1] 100% CNT50
    \\\2:J  P[2] 100% FINE
    [End]
```

7 To view details of an entry, select the entry and press F3, DETAIL. You will see a screen similar to the following:



```
Log book
    Operation        ( 50.0 k)      5/790
    SRVO-001 Operation panel E-stop


    SERVO                              00110110
    24-JUN-00 14:17 :36
```

8 To clear the log, press F5, CLEAR, then press F4, YES.

Saving Log Book data as a Text File

Log Book data is saved as text file LOGBOOK.LS. There are two ways to save data:

- At the FILE menu, press F4, BACKUP, and select “Error Log”.
  LOGBOOK.LS is saved along with the error log files.
- In the Log Book menu, press [FCTN] key and select SAVE.
  LOGBOOK.LS is saved to the selected device.

See the following example of a LOGBOOK.LS file.

```
Operation
============================================================
*SHIFT,F5(TOUCHUP) is pressed, line 2/3
00/06/02 14:17:36
--Screen image-------------------------------
PNS0001\\\\\\\\\\\\\\\\\\\\\\\JOINT\\10\%
   1 J P[1] 100% CNT59
   2 J P[2] 100% FINE
 [END]
 POINT                        TOUCHUP>
============================================================
'+Y^(J6)' is pressed
00/06/02 14:17:20
============================================================
SHIFT+FWD is pressed
00/06/02 14:17:12
============================================================
Select 'YES' in 'The cursor is on a different' window
00/06/02 14:16:40
--Screen image-------------------------------
PNS0001\\\\\\\\\\\\\\\\\\\\\\\JOINT\\10\%
     +------------------------------+
     |The cursor is on a different  |
     |line from where the program   |
   1 |PAUSED [1].                   |
   2 |Are you sure you want to run  |
 [END|from this line ?              |
     |                              |
     |   \\\\YES\\\\         NO     |
     |                              |
     +------------------------------+
 POINT                        TOUCHUP>
============================================================
SHIFT+FWD is pressed
00/06/02 14:16:30
============================================================
SHIFT+FWD is pressed
00/06/02 14:16:28
============================================================
```

Filtering Alarms and Screens

You can set system variables to filter specific alarms and screens.

By default, Log Book filters out warning errors. You can filter errors to be logged by severity, type, and item number.

Table 12.3 (a) System variables used for filtering

System Variable Name	Type	Default	Range	Description
$LOG_ER_SEV	INTEGER	6	-2147483648 2147483647	Filter by severity of error. When a bit in this is True, the corresponding errors are logged. • Bit 0 (1): Log warning errors. • Bit 1 (2): Log pause errors. • Bit 2 (4): Log abort errors. Ex. 6 = pause and abort errors are logged. Priority of this is lower than $LOG_ER_TYP and $LOG_ER_ITM.
$LOG_ER_TYP[1-n]	INTEGER	0	-2147483648 2147483647	Filter by type of error. If a positive value is specified, alarms of the specified type are logged. Ex. 11 = SRVO alarms are logged. If a negative value is specified, alarms of the specified type are not logged. Ex. -11 = SRVO alarms are not logged. Priority of this is higher than $LOG_ER_SEV but lower than $LOG_ER_ITM.
$LOG_ER_ITM[1-n]	INTEGER	0	-2147483648 2147483647	Filter by individual error. If a positive value is specified, the specified alarm is logged. Ex. 11001 = SRVO-001 is logged. If a negative value is specified, the specified alarm is not logged. Ex. -11001 = SRVO-001 is not logged. Priority of this is higher than $LOG_ER_SEV and $LOG_ER_TYP.

Control error filtering by setting the following system variables.

- $LOG_ER_ITM[1-n]
- $LOG_ER_SEV
- $LOG_ER_TYP[1-n]

The priority of the settings is: $LOG_ER_SEV < $LOG_ER_TYP < $LOG_ER_ITM
See the following for an example of error filtering:

```
SRVO-001 "Operator panel E-stop"     Severity=PAUSE  Type=11 Item=11001
SRVO-038 "Pulse mismatch"            Severity=ABORT  Type=11 Item=11038
TPIF-104 "Teach pendant is disabled" Severity=WARN   Type=9  Item=9104
```

Table 12.3 (b) Alarm filtering example

$LOG_ER_SEV	$LOG_ER_TYP	$LOG_ER_ITM	SRVO-001	SRVO-038	TPIF-104
6	0, 0, ..	0 ,0, ..	Logged	Logged	Not logged
6	0, 0, ..	9104, 0, ..	Logged	Logged	Logged
6	0, 0, ..	9104, -11001, ..	Not logged	Logged	Logged
7	-11, 0, ..	11001, 0, ..	Logged	Not logged	Logged
0	11, 9, ..	-11001	Not logged	Logged	Logged

Screen filtering enables Logbook to log UIF events that occur in the screens you specify. Screen filtering supports the following events (UIF events).

- 'x' is entered.
- 'x' is pressed.
- 'x' is selected.
- 'x' is selected in 'y' menu.
- 'x' is selected in 'y' window.
- 'x' is selected in MENU.
- 'x' is selected in FCTN.
- JOG menu TOOL 1 etc.

NOTE

TP’x’ ON/OFF is not supported.

By default, screen filtering is disabled. Screen filtering is enabled/disabled by $LOGBOOK.$SCRN_FL.

- If $LOGBOOK.$SCRN_NO_ENT is TRUE, events in registered screens are recorded. If FALSE, events in NOT registered screens are recorded.
- To register screens, softpart ID and screen ID must be set to $LOG_SCRN_FL[ ].$SP_ID and $LOG_SCRN[ ].$SCRN_ID. By default, no screen is registered for filter.

Table 12.3 (c) System variables for screen filtering

System Variable Name	Type	Default	Range	Description
$LOGBOOK.$SCRN_FL	BOOLEAN	FALSE	FALSE/TRUE	Screen filter is disabled/enabled.
$LOGBOOK.$SCRN_NO_ENT	BOOLEAN	TRUE	FALSE/TRUE	TRUE: Events on registered screens are not recorded. Events on other screens are recorded. FALSE: Events on registered screens are recorded Events on other screens are not recorded.
$LOGBOOK.$NUM_SCRN_FL	Integer	20	1,200	Number of $LOG_SCRN_FL
$LOG_SCRN_FL[ ].$SCRN_ID	ULONG	0	0,4294967295	Screen ID of screen for filter
$LOG_SCRN_FL[ ].$SP_ID	ULONG	0	0,4294967295	Softpart ID of screen for filter

To filter screens so that passwords are not recorded, set the following:

```
$LOGBOOK.$SCRN_FL = TRUE
$LOG_SCRN_FL[1].$SP_ID = 935
$LOG_SCRN_FL[1].$SCRN_ID = 1
```

12.4 EXTENDED ALARM LOG

12.4.1 Setup

This example name BOOK3 as “ALARM” and use it for only for record of alarms. This configuration uses PERM memory of 500kbytes. The number of alarms that can be stored in the book depends on contents of alarms. Suppose user alarm with undefined alarm message occurred. The book can record the alarm about 5000 times.

NOTE

Because size of book is large, please check the rest of PERM memory before using this setting. The rest of PERM memory should be more than 150KB after the book is established. If the book uses 500KB, the rest of PERM memory should be more than 650KB (500KB + 150KB).

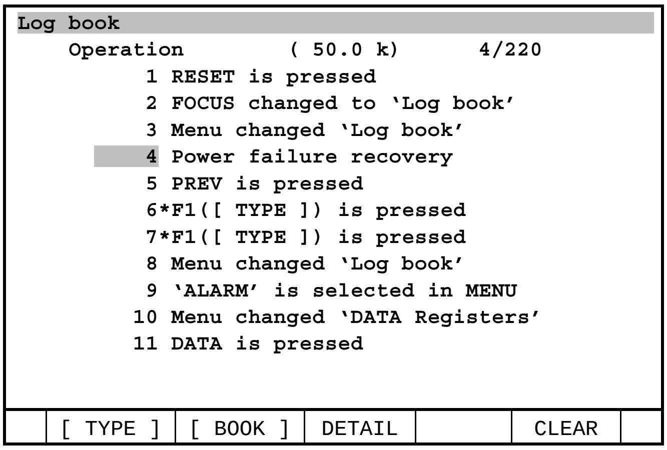
Table 12.4.1 System variables

System variable	Type	Default value	Value	Comment
$LOG_BUFF[3]. $TITLE	String	‘’	ALARM	Title of book
$LOG_BUFF[3]. $SIZE	Integer	0	500	Size of book is 500kbytes. The number of alarms that this book can store depends on contents of alarms.
$LOG_BUFF[3]. $MEM_TYPE	Integer	0	0	Buffer of this book is made on permanent pool. Cycle power doesn’t clear records.
$LOG_BUFF[3]. $VISIBLE	Boolean	TRUE	TRUE	The book of “ALARM” is displayed in F2 ([BOOK]) pull up menu. Please note that this book is not always displayed as the 3rd item on the pull-up menu.
$LOGBOOK. $LOG_ER	Integer	1	3	Alarms are recorded in book3.
$LOG_ER_SEV	Integer	6	7	Filter by severity of error. When a bit in this is TRUE, the corresponding errors are logged. • Bit 0 (1): Log warning errors. • Bit 1 (2): Log pause errors. • Bit 2 (4): Log abort errors. Ex. 7 = warn, pause and abort errors are logged (All alarms are recorded).

Cycle power is needed to enable these changes.

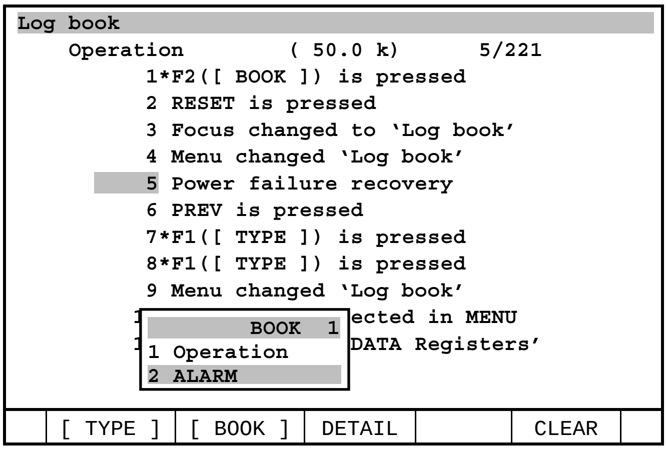
12.4.2 How to Display Alarm Log

1) Press [MENU] key.
Select ALARM.
Press F1 and select “Log book”. Logbook screen will be displayed.



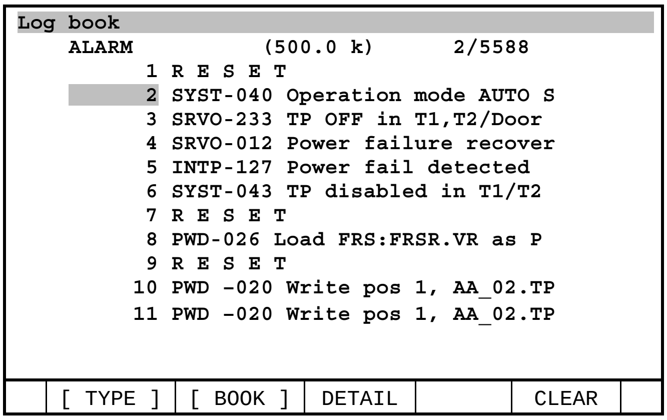
```
Log book
    Operation        ( 50.0 k)      4/220
          1 RESET is pressed
          2 FOCUS changed to ‘Log book’
          3 Menu changed ‘Log book’
          4 Power failure recovery
          5 PREV is pressed
          6*F1([ TYPE ]) is pressed
          7*F1([ TYPE ]) is pressed
          8 Menu changed ‘Log book’
          9 ‘ALARM’ is selected in MENU
         10 Menu changed ‘DATA Registers’
         11 DATA is pressed

   [ TYPE ] [ BOOK ]  DETAIL          CLEAR
```

2) Press F2 and select “ALARM”.



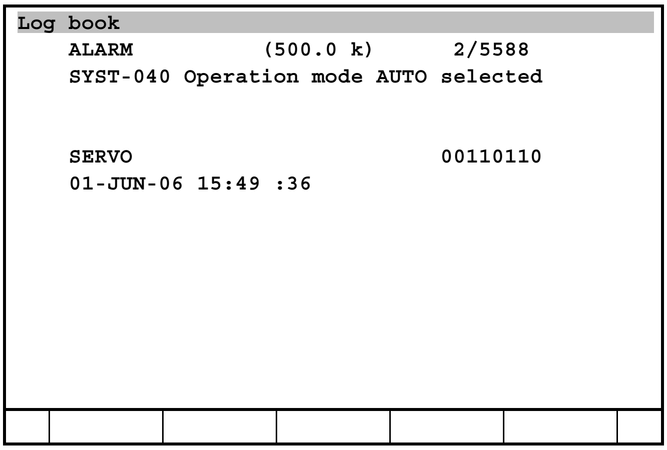
```
Log book
    Operation        ( 50.0 k)      5/221
          1*F2([ BOOK ]) is pressed
          2 RESET is pressed
          3 Focus changed to 'Log book'
          4 Menu changed 'Log book'
          5 Power failure recovery
          6 PREV is pressed
          7*F1([ TYPE ]) is pressed
          8*F1([ TYPE ]) is pressed
          9 Menu changed 'Log book'
         1                ected in MENU
         1                DATA Registers'
           BOOK  1
          1 Operation
          2 ALARM

   [ TYPE ]  [ BOOK ]  DETAIL         CLEAR
```

3) Contents of book “ALARM”, which is made by setting of this example, is displayed.



```
Log book
    ALARM          (500.0 k)      2/5588
          1 R E S E T
          2 SYST-040 Operation mode AUTO S
          3 SRVO-233 TP OFF in T1,T2/Door
          4 SRVO-012 Power failure recover
          5 INTP-127 Power fail detected
          6 SYST-043 TP disabled in T1/T2
          7 R E S E T
          8 PWD-026 Load FRS:FRSR.VR as P
          9 R E S E T
         10 PWD -020 Write pos 1, AA_02.TP
         11 PWD -020 Write pos 1, AA_02.TP

   [ TYPE ]  [ BOOK ]  DETAIL         CLEAR
```

4) F3 shows detail screen.



```
Log book
    ALARM          (500.0 k)      2/5588
    SYST-040 Operation mode AUTO selected


    SERVO                        00110110
    01-JUN-06 15:49 :36
```

13 PROGRAM TOOLBOX

The Program Toolbox has the following features. For except Spot Tool+, to use this function, Program Toolbox option (A05B-2xxx-R598) is required.

- Soft limit setting

13.1 SOFT LIMIT SETTING

Using the soft limit setting option enables the software axis limits for a robot to be determined automatically; so the hard stop locations for the J1 axis can be determined. Setting up axis limits without using the soft limit setting option requires executing all programs step by step, recording the maximum value for each angle before the axis joint limits can be set up manually. It also requires jogging the robot to the position limits specific to the J1 axis and determining the best hardware stop position for the J1 axis manually.

The soft limit setting option reads all of the programmed positions on the robot automatically and determines the maximum and minimum taught joint angles used in all programs. It then uses the information to set up the specified joint limits automatically while taking a user-specified limit buffer into account. The soft limit setting option also reports the appropriate locations for the J1 axis hard stop according to the same maximum taught joint angle for all programs.

By default, software axis limits are set up only for the J1, J2, and J3 axes. If necessary, they can be set up for other axes.

A limit buffer is added to the detected limits to allow tolerance for motion between positions when a program is executed. When the robot moves from one position to another, the motion of the robot between the positions might get out of the axis limits. The limit buffer is applied to the detected maximum and minimum taught points to ease the axis limits so that a joint limit error will not occur on the robot motion between the taught points.

A limit buffer can be set anywhere between 0 and 50 degrees. A limit buffer of 10 degrees is set up by default. In many cases, this limit buffer degree provides adequate ease. If many limit errors occur during program execution after the soft limit setting option is used, increase the limit buffer and try again.

Before limits are set up, axes to which the limits are to be applied must be selected and, if necessary, the limit buffer value must be re-set. Factory-set limits can be restored if newly set values become unnecessary.

Using the limit setting option

Table 13.1 lists each soft limit setup item. Use Procedure 13-1 to calculate mounting locations for limit blocks on the robot flange.

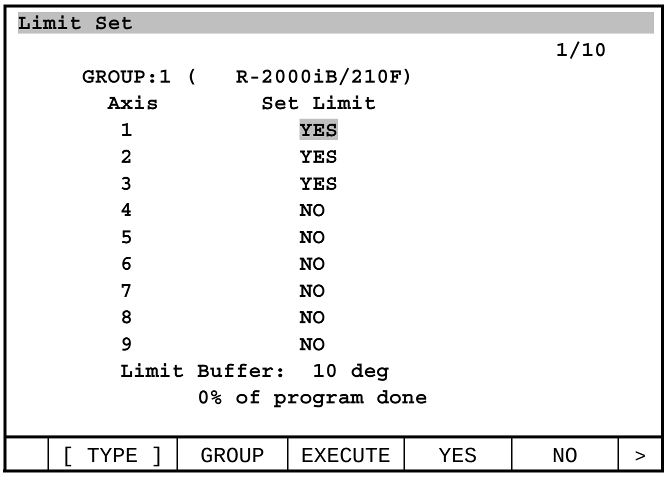
Table 13.1 Toolbox soft limit setup menu items

Item	Description
Axis	This item is the number assigned to the axis for which limits can be set up.
Set Limit	This item indicates whether a limit has been set up.

Procedure 13-1 Using the soft limit setting option to set up software axis limits

Step

1 Press [MENU] key.
2 Select UTILITIES.
3 Press F1, [TYPE].
4 Select Limit Set. The following menu will be displayed.



```
Limit Set
                                            1/10
     GROUP:1 (   R-2000iB/210F)
       Axis        Set Limit
        1             YES
        2             YES
        3             YES
        4             NO
        5             NO
        6             NO
        7             NO
        8             NO
        9             NO
        Limit Buffer:  10 deg
              0% of program done
```

5 To select an axis for limit setting, follow the steps below:

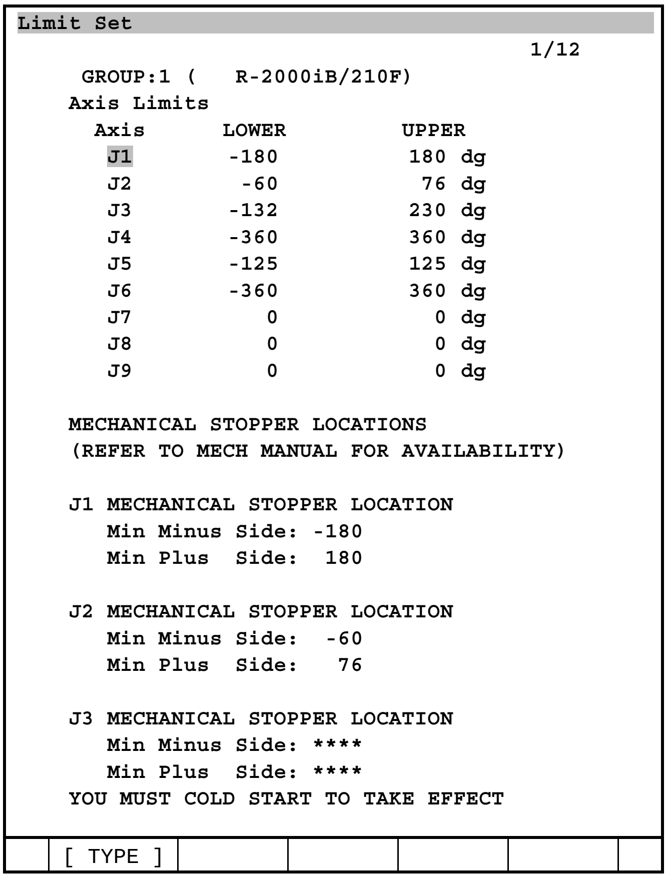
  a. Move the cursor to the number assigned to a desired axis.

  b. Press F4, Yes to select the axis. If F5, No is pressed, no limit is set up for the axis,

6 To specify a limit buffer, move the cursor to Limits buffer, enter a degree, and press [ENTER] key.

7 To set up the axis limits, press F3, EXECUTE.

When limit setting is completed, the following information will be displayed on the screen.



```
Limit Set
                                          1/12
     GROUP:1 (   R-2000iB/210F)
    Axis Limits
      Axis      LOWER         UPPER
       J1        -180          180 dg
       J2         -60           76 dg
       J3        -132          230 dg
       J4        -360          360 dg
       J5        -125          125 dg
       J6        -360          360 dg
       J7           0            0 dg
       J8           0            0 dg
       J9           0            0 dg

    MECHANICAL STOPPER LOCATIONS
    (REFER TO MECH MANUAL FOR AVAILABILITY)

    J1 MECHANICAL STOPPER LOCATION
       Min Minus Side: -180
       Min Plus  Side:  180

    J2 MECHANICAL STOPPER LOCATION
       Min Minus Side:  -60
       Min Plus  Side:   76

    J3 MECHANICAL STOPPER LOCATION
       Min Minus Side: ****
       Min Plus  Side: ****
    YOU MUST COLD START TO TAKE EFFECT
```

The displayed mechanical stop numbers indicate the locations of the limit blocks on the J1, J2, and J3 axes. Mounting the limit block on the J1 axis requires aligning the hole at the center of the limit block with the reported J1 axis mechanical stop hole on the base of the robot.

Additional information

The displayed mechanical stop numbers indicate the positions of the center bolt hole on each mechanical stop.

Note: The current version does not support the mounting positions of mechanical stops.
Additional information
The following operation is not supported on some robot models.

8 To reset the axis limits to their factory settings, press F1, DEFAULT.
Additional information
Enabling new software axis limit settings always requires cold-starting the controller.

9 After axis limit setting is completed, enable the new software axis limit settings by cold-starting the controller according to the following steps.

a. If the controller is already on, turn it off.
b. On the teach pendant, press and hold down the SHIFT and RESET keys.
c. While still holding down the teach pendant keys, turn the power on.
d. After the teach pendant has displayed its menu, release the teach pendant keys.

Additional information
If many joint limit errors occur during program execution, increase the limit buffer and re-run the program.

14 ADVANCED CONSTANT PATH

ADV-CP Path CTRL allows you to use Linear Distance and Corner Region for adjusting the path. ADV-Max Speed CTRL allows you to use Process Speed and Max Speed for adjusting the speed of a Robot.

ADV-CP Path CTRL (R806)
- Linear Distance
- Corner Region

ADV-Max Speed CTRL (R805)
- Process Speed
- Max Speed

It requires that the Constant Path function be enabled.

14.1 LINEAR DISTANCE

Linear Distance is a motion feature that is useful for material handling applications such as palletizing. Fig.14.2(a) shows a typical pick and place application. If a robot starts at P1 and goes through P2 and picks up a part at P3 and then goes through P2 to P4 and then to P5 where it places the part and all termination types are FINE or CNT0, then the path would be as shown.

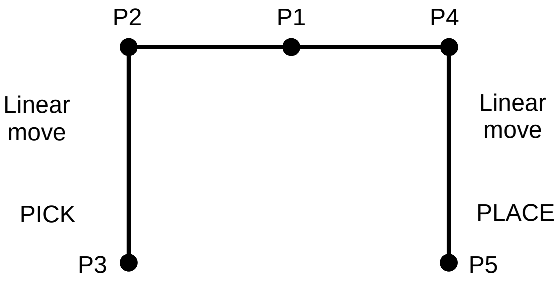
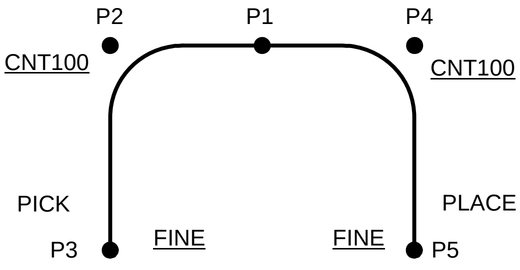


Fig. 14.1 (a) PICK and PLACE application

Typically, however, FINE and CNT0 is used only to reach P3 for PICK and to PLACE at P5. For all other motions, high CNT values are used. For instance, if CNT100 is used, the actual path might look like Fig.14.2(b).

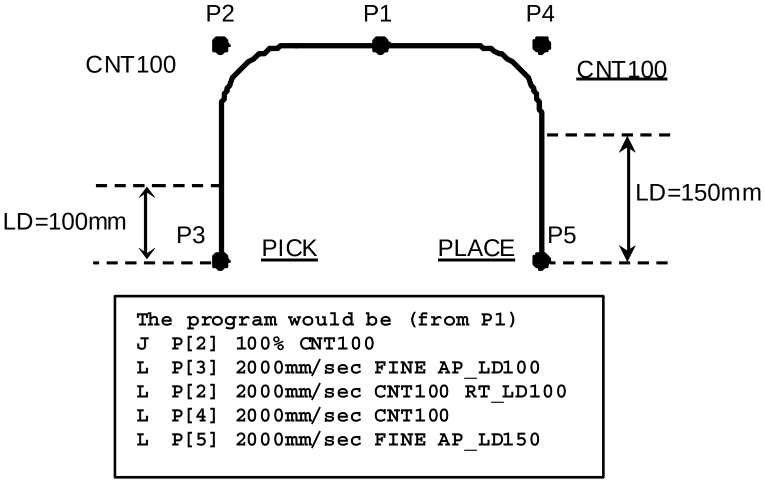


```
Robot is at P1
J  P[2] 100% CNT100
L  P[3] 2000mm/sec FINE
L  P[2] 2000mm/sec CNT100
L  P[4] 2000mm/sec CNT100
L  P[5] 2000mm/sec FINE
```

Fig. 14.1 (b) PICK and PLACE application with CNT100

This kind of a path will give you better cycle time. However, you can not specify the linear part of the path from P2 to P3 or from P4 to P5 directly. When you want to adjust the linear part of the path, you need to adjust values of CNT.

With Linear Distance, you do not have to guess and experiment. If you want the last 100mm before pick and the last 150mm above place to always be straight above P3 and P5 respectively, you can use Linear Distance for specifying these amounts. Refer to Fig. 14.1 (c).



```
The program would be (from P1)
J  P[2]  100% CNT100
L  P[3]  2000mm/sec FINE AP_LD100
L  P[2]  2000mm/sec CNT100 RT_LD100
L  P[4]  2000mm/sec CNT100
L  P[5]  2000mm/sec FINE AP_LD150
```

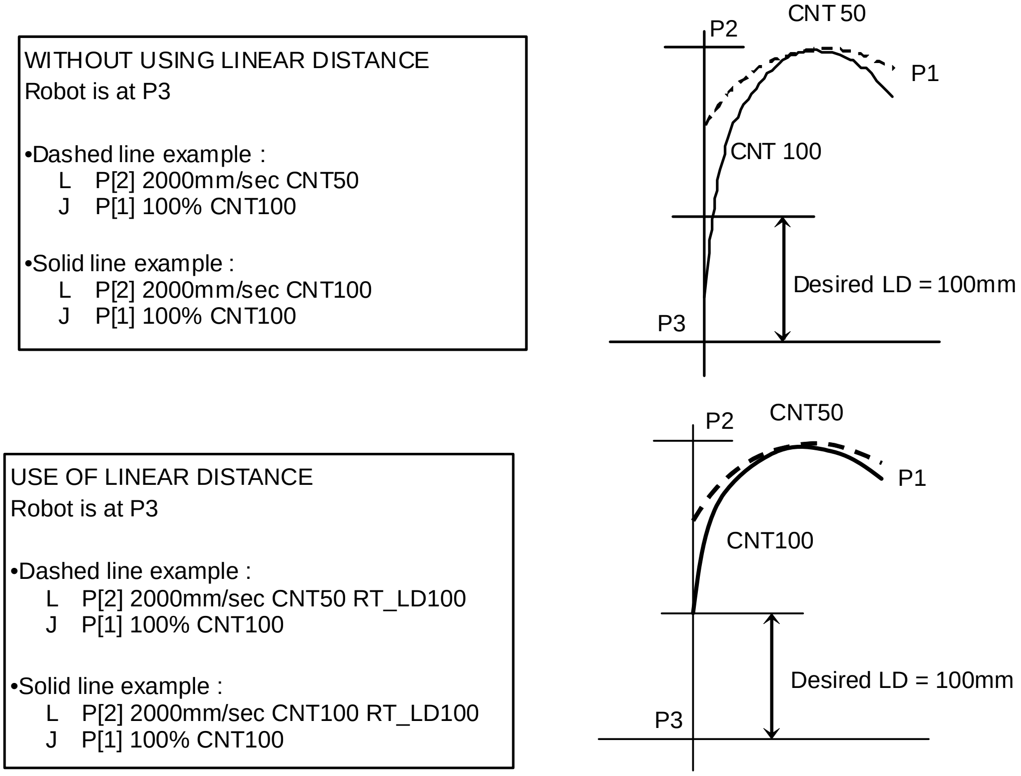
Fig. 14.1 (c) Adjusting P3 and P5 with linear distance

14.1.1 How to Use

Linear Distance allows you to use two additional motion instructions:

RT_LD: Retract Linear Distance is used for specifying the desired linear distance from the start position to the start of the corner.

AP_LD: Approach Linear Distance is used for specifying the desired linear distance from the end of the corner to the destination position.



WITHOUT USING LINEAR DISTANCE
Robot is at P3

•Dashed line example :
L P[2] 2000mm/sec CNT50
J P[1] 100% CNT100

•Solid line example :
L P[2] 2000mm/sec CNT100
J P[1] 100% CNT100

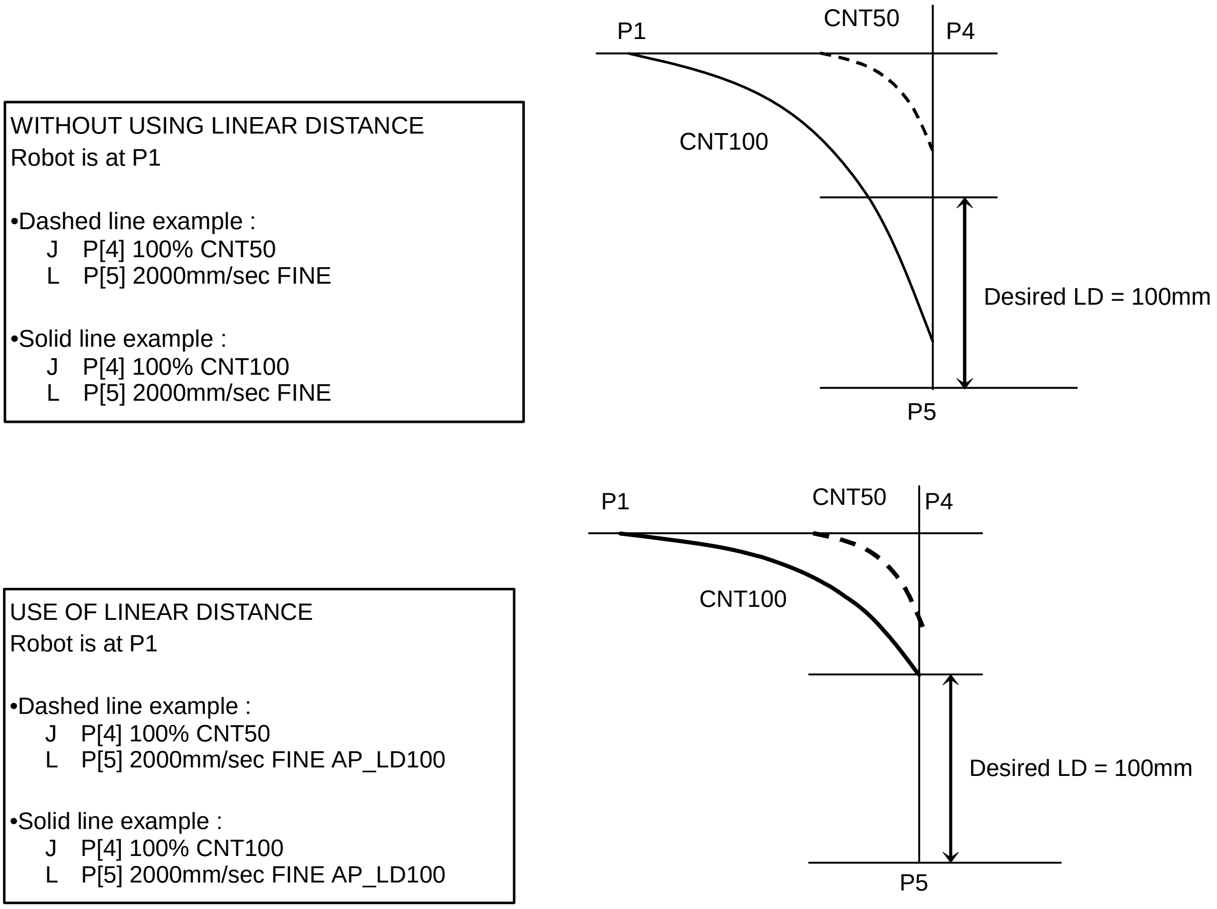
USE OF LINEAR DISTANCE
Robot is at P3

•Dashed line example :
L P[2] 2000mm/sec CNT50 RT_LD100
J P[1] 100% CNT100

•Solid line example :
L P[2] 2000mm/sec CNT100 RT_LD100
J P[1] 100% CNT100

Fig. 14.1.1 (a) RT_LD: Effect of CNT value

The RT_LD value affects the corner of P3-P2-P1 in Fig. 14.1.1(a). The higher the value of RT_LD, the shorter the distance between the path and P2 will be. When the RT_LD value is greater than or equal to the distance between P3 to P2, the motion will automatically become like FINE regardless of the CNT value you specify.



```
WITHOUT USING LINEAR DISTANCE
Robot is at P1

•Dashed line example :
    J   P[4] 100% CNT50
    L   P[5] 2000mm/sec FINE

•Solid line example :
    J   P[4] 100% CNT100
    L   P[5] 2000mm/sec FINE
```

```
USE OF LINEAR DISTANCE
Robot is at P1

•Dashed line example :
    J   P[4] 100% CNT50
    L   P[5] 2000mm/sec FINE AP_LD100

•Solid line example :
    J   P[4] 100% CNT100
    L   P[5] 2000mm/sec FINE AP_LD100
```

Fig. 14.1.1 (b) Place motion: two possible traces

The AP_LD value affects the corner of P1-P4-P5 in Fig. 14.1.1(b). The higher the value of AP_LD, the shorter the distance between the path and P4 will be. When the AP_LD value is greater than or equal to the distance between P4 to P5, the preceding motion (motion from P1 to P4) will become like FINE regardless of the CNT value for that move.

14.1.2 Limitations

- Linear Distance can not match the specified value exactly. However, it will never provide linear distance less than what you specified.
- Linear Distance can be used with only linear motion type.
- Linear Distance can not be used with Independent Axis or Positioner. If you use linear distance instructions, the instructions are disabled.
- When multiple group motion is used, the motion will be synchronized. However, if more than one group has linear distance enabled, all the groups will have linear distance satisfied.
- The local condition trigger time might be different than without Linear Distance. However the timing is repeatable.
- Linear Distance can not be used with the following options.
  - Max Speed
  - Weaving
  - Continuous turn

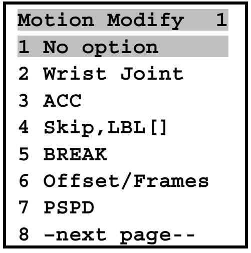
  - Robot Link
  - TCP speed output function
- Linear Distance can be used with Coordinated Motion.

14.1.3 Procedure to Use

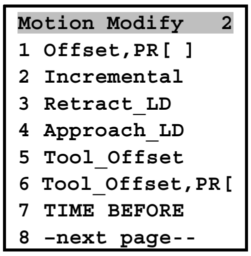
Procedure 14-1 Procedure to use

Step

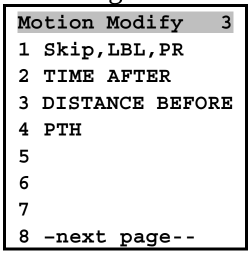
1 Press [SELECT] key.
2 Move the cursor to the name of the program you want to modify and press ENTER.
3 Move the cursor to the empty space at the end of the linear motion instruction that you want to modify and press F4, [CHOICE]. You will see menus similar to the following.



```
Motion Modify   1
1 No option
2 Wrist Joint
3 ACC
4 Skip,LBL[]
5 BREAK
6 Offset/Frames
7 PSPD
8 -next page--
```



```
Motion Modify    2
1 Offset,PR[ ]
2 Incremental
3 Retract_LD
4 Approach_LD
5 Tool_Offset
6 Tool_Offset,PR[
7 TIME BEFORE
8 -next page--
```



```
Motion Modify    3
1 Skip,LBL,PR
2 TIME AFTER
3 DISTANCE BEFORE
4 PTH
5
6
7
8 -next page--
```

4 If you want to specify the linear distance from the start position to the start of the corner, please select Retract_LD.
If you want to specify the linear distance from the end of the corner to the destination position, please select Approach_LD.
5 Type the number of millimeters that you want the tool center point (TCP) to approach or retract using Linear Distance.

> ⚠ **NOTE**
> The default value is “direct” which means that the value is a specific number in millimeters. To use a value stored in a register, press F3, INDIRECT, and type the register number.

14.2 CORNER REGION

L P[2] 100 mm/sec CRy

Corner Region (CR) is an optional termination type that can be used to adjust the corner rounding for Cartesian motions. When you use the CR termination type, you must specify the corner region value, y, (in millimeters). In addition to that, when you use the CR termination type, the speed at the corner keeps constant as far as possible.

Corner Region value is the distance from the starting of a corner path to the taught position, as shown in the following figure. When CR~~y~~ is specified, the tool center point path will maintain the corner path within the specified region, meaning that the actual distance from the starting/ending of a corner path to the taught position is less than or equal to the specified corner region value.

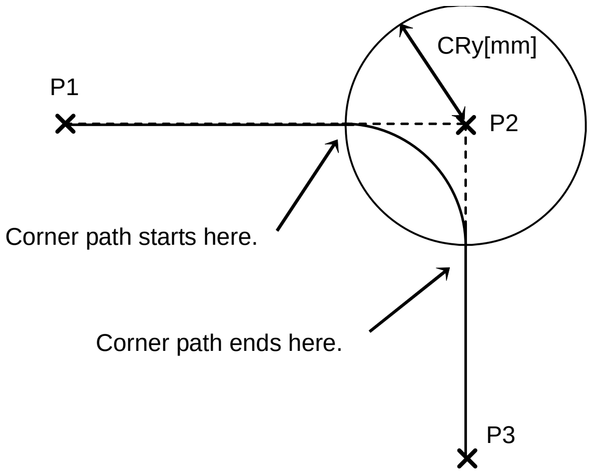


Fig. 14.2 Corner path

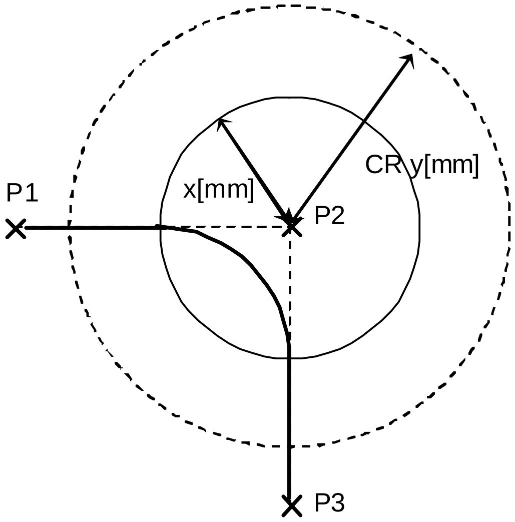
When you set corner region, use the following guidelines:

- Specify the corner region in millimeters.
- Corner region value can range in value from 0 mm to 1000 mm.
- The smaller the corner region value, the closer the robot will get to the taught position, and the less the corner rounding.
- With a larger corner region value specified, the robot will not get as close to the position and the more corner rounding.

14.2.1 Limitation of the Specifiable CR Value

If the specified corner region value, y, is greater than half distance between P1 and P2 (P2 and P3), then the actual value used is limited to half distance between P1 and P2 (P2 and P3) as shown below.

CR y[mm] > half distance between P1-P2 (P2-P3)
Actual CR x=half distance between P1-P2 (P2-P3)



Unequal distance case
Actual CR = x1, x2 (half distance between P1-P2 (P2-P3))

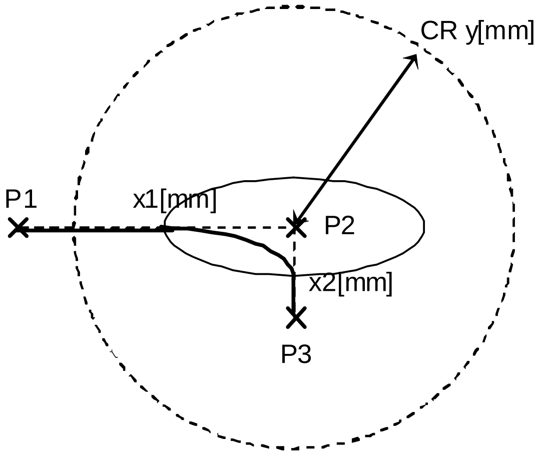


Fig. 14.2.1 Limitation of the specifiable CR value

You must be careful about the half distance rule stated above. Please keep in mind that because of the half distance rule, with large specified corner region value, corner path may still be close to the taught position when the distance between continuous taught points is short. Use the following guidelines when you teach a path:

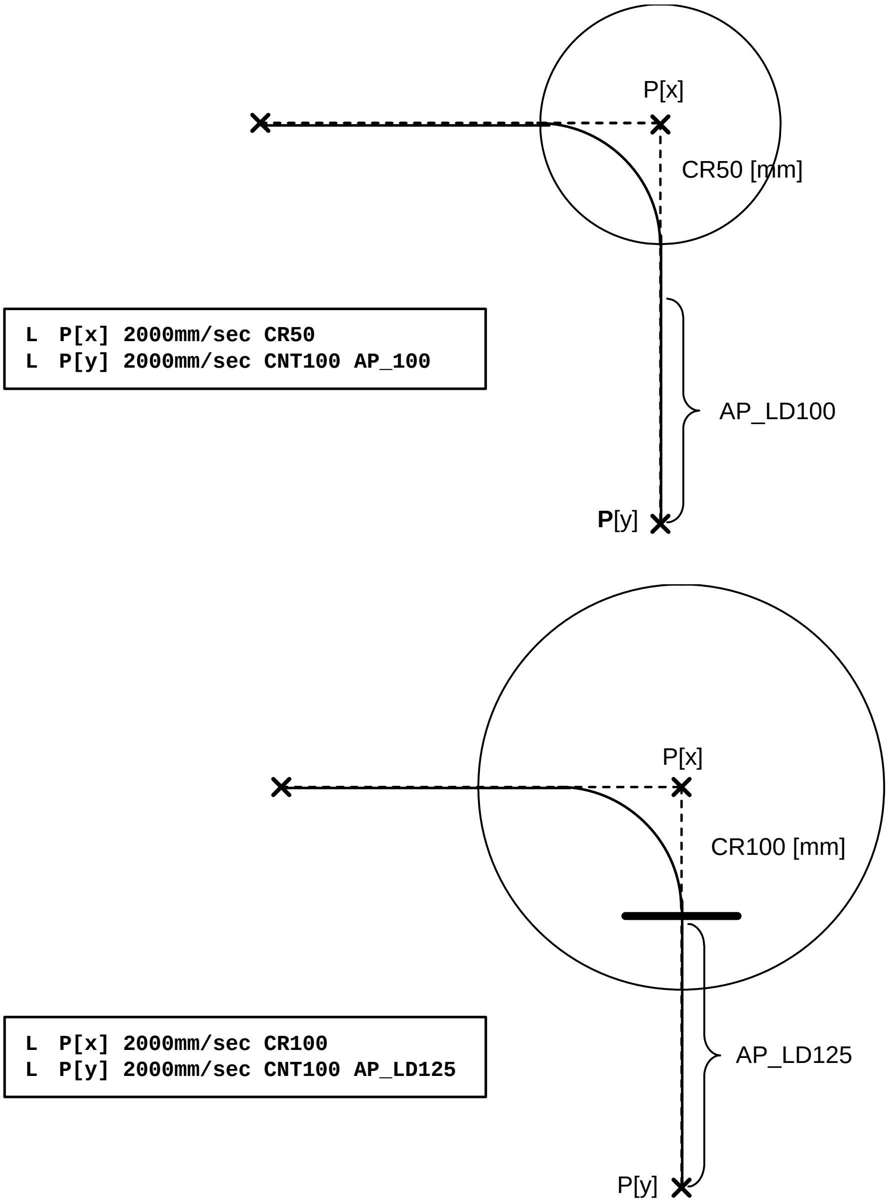
- Minimize the number of taught positions.
- Reteach positions using the CR termination type to fit the path instead of adding positions.

14.2.2 Limitations

- CR termination type can not be used with JOINT motion type.
- CR termination type can not be used with Independent Axis or Positioner. When Independent Axis or Positioner move with Robot, it is possible to use CR termination type. However, in this case, CR takes effect for only Robot motion. Independent Axis or Positioner synchronizes with Robot.
- When multiple Robots move with multi group, the motion of robots is synchronized. In this case, all robot move so that the corner region is within the specified value.
- f the angle between the motion and the next motion is almost 180 degrees, CR termination type does not work and the path becomes the same as CNT100.
- The speed at the corner keeps constant as far as possible. However, when there is an instruction like the following, the speed decreases because it is impossible to read ahead of the motion.
  - 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.
  - Motion Statements with the indirect addressing with the register.
  - Branch instruction (IF or SELECT JMP LBL[*]) is used before the next motion statement.
  - WAIT instruction with TIMEOUT LBL[*] is used before the next motion statement.
  - Frame instructions are used before the next motion statement.
- CR termination type can be used with the following options.
  - Remote TCP
  - Line Tracking

    During the transition (between Remote TCP and non-Remote TCP or between Tracking and non-Tracking), corner can not be defined because a frame is changed. Therefore the corner region can not be specified by CR but the corner path is connected continuously.
  - Linear Distance

    If you specify both Linear Distance and CR termination type, then Linear Distance has preference over CR, as shown following figures.



```
L  P[x] 2000mm/sec CR50
L  P[y] 2000mm/sec CNT100 AP_100
```

```
L  P[x] 2000mm/sec CR100
L  P[y] 2000mm/sec CNT100 AP_LD125
```

Fig. 14.2.2 Corner Path determined by CRy if Linear Dist is satisfied

14.3 PROCESS SPEED

PSPD xxx

Process speed is a motion option control feature that allows you to adjust robot speed to be faster or slower along a given path (if applicable, the path would be maintained the same regardless of xxx), where xxx is an integer you specify. The larger the value of xxx is, the faster the robot will move along the given path.

Process speed is useful for applications with continuous path motion that don’t normally use maximum program speed; for example, sealing and waterjet cutting. Typically, the process controls the program speed: how fast the sealing gun can dispense, and how fast the waterjet can cut.

For these applications, teach the desired path using normal methods, tweaking taught position, speed, term type and ACC.

After the path is taught, if you want to adjust its process speed from the nominal taught value, but do not want to change the path, you can use the Process Speed feature.

Add this motion option to the range of motion lines where adjustment is required.

- PSPD 100 is equivalent to the default cases without PSPD option.

- PSPD greater than 100 means faster process speed, while maintaining the same path.
- PSPD less than 100 means slower process speed, while maintaining the same path.
- You can still change other fields in the motion for further tweaking, but the same original rules apply; that is, the path will change. This allows you to adjust the path easily, even though PSPD is used.
- For PSPD greater than 100, the system internally limits the achievable (but higher) process speed, based on the jerk/acceleration margin available from the default case.

Be careful to use the PSPD option to reduce cycle time while maintaining the same path since jerk/acceleration value will be higher. An example is palletizing, where additional factors such as vibration, duty cycle, reducer life, and so forth, affect cycle time.

• Path Teaching
J P[1] 100% FINE
L P[2] 500mm/sec CNT100 ACC120
L P[3] 1000mm/sec CNT0 ACC130

• Cycle Time Optimization
J P[1] 100% FINE
L P[2] 500mm/sec CNT100 ACC120 **PSPD110**
L P[3] 1000mm/sec CNT0 ACC130 **PSPD110**

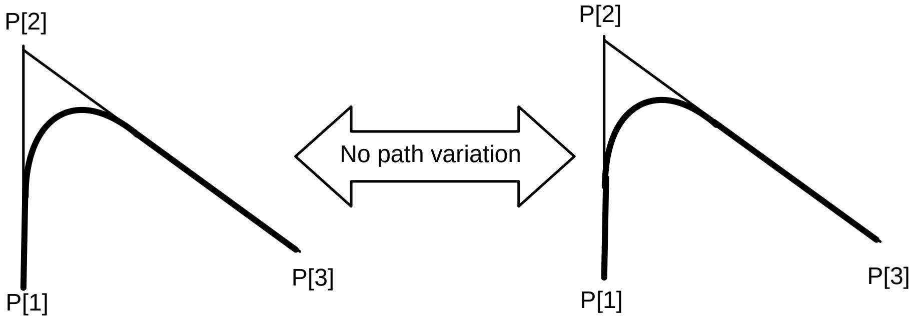


Fig. 14.3 Process speed tweaking

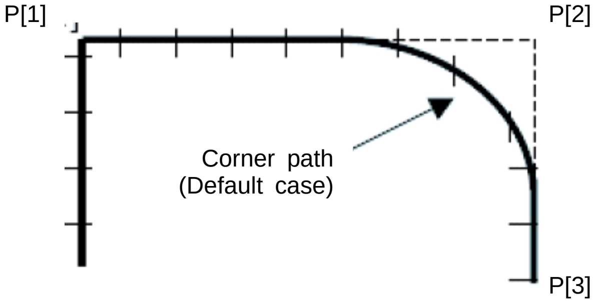
⚠ **CAUTION**
Process Speed can cause jerky motion if applied too aggressively. To avoid jerky motion, use a reduced speed.

PSPDxxx can be added to any selective motion line in a TP program, and is applicable to all motion types. For examples, see the following:

- Case 1 : *** = 100, motion behavior is exactly the same as 100% speed override in the default case, as though there were no PSPD100.
- Case 2 : *** > 100, the speed will be faster than 100% speed override in the default case.
- Case 3 : *** < 100, the speed will be slower than 100% speed override in the default case.

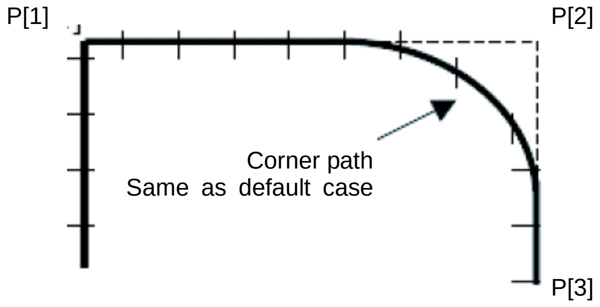
- Default case :

J P[1] 50% FINE
L P[2] 500mm/sec CNT100
L P[3] 500mm/sec CNT0



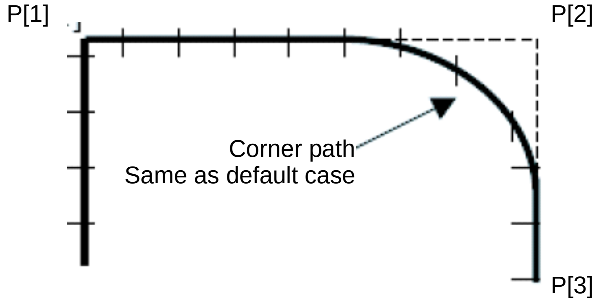
- Faster motion case (path is the same, cycle time is shorter):

```
J P[1] 50% FINE PSPD110
L P[2] 500mm/sec CNT100 PSPD110
L P[3] 500mm/sec CNT0 PSPD110
```



- Slower motion case (path is the same, cycle time is longer):

```
J P[1] 50% FINE PSPD50
L P[2] 500mm/sec CNT100 PSPD50
L P[3] 500mm/sec CNT0 PSPD50
```



⚠ **NOTE**

The system will internally limit the speed override such that the resulting motion performance is within mechanical capabilities. As a result, a large value of xxx may not take effect in some cases and the actual speed override may be smaller than the specified value.

14.3.1 Limitations

- Under T1 mode, PSPDxxx (with xxx>100) will not take effect.
- PSPDxxx (with xxx>100) might not take effect for the motion line that has max_speed as programmed speed in a TP program.
- The PSPD option does NOT support TCP speed prediction function (TCPP). That is, for the motions with PSPD option, TCPP might not result in correct results.
- With large PSPD value or very short segments, the actual corner path might deviate from the one without PSPD option.

14.4 MAX SPEED

L P[1] max_speed CNT100

In some applications, the desired speed is the maximum speed that the robot can deliver. For joint motion moves, the system delivers the maximum capability of the robot; that is, one of the axes reaches its maximum speed. For linear moves, the system delivers the speed that is specified in the teach pendant instruction. However, the maximum linear speed of 2000mm/sec imposes a limit on the capability of the motor to reach higher speeds. The robot can move faster than the speed specified in the motion instruction.

The max speed option allows you to specify a linear motion that will use the maximum speed capability of the robot. It improves cycle times in Load/Unload applications by speeding up long linear motions. When this option is loaded, the choice of max_speed will be displayed in the speed field of the teach pendant motion instruction for a linear motion. The max_speed option affects only the motions for which the speed is specified as max_speed.

- If you change the motion type from Linear to Joint, the speed field will change to 100%.
- When the speed field changes from max_speed to another choice, the speed value will return to the original speed value.

> ⚠ **WARNING**
> When you specify max_speed, the robot will run at high speed. Be sure any loose parts are firmly attached and that the workpiece is secured.
> Otherwise, you could injure personnel or damage equipment.

14.4.1 Limitations

- If unsupported options are used, max_speed will be disabled automatically. No warning or error message will be displayed. This option does not support the following:
  - Any tracking option, such as line tracking, TAST, Coordinated motion, and so forth.
  - Multiple group motion
  - RTCP function
- If you run a program with an override speed different than 100%, the system will drive the robot such that one of its axes will reach the override value of its maximum joint speed.
- The local condition trigger time might have some variation.
- If the path becomes too aggressive, you might need to use ACC to smooth it.
- If you are using Dry Run, max speed will be disabled and the speed specified in dry run will be used.
- If you are using Org path resume, max speed will be disabled for the motion line that is resumed.
- If T1 is selected, the T1 speed will be used.
- In single step mode (FWD/BWD) max speed will be disabled and the maximum speed value will be used.
- Max speed will be disabled automatically for a circular motion.
- The max speed option will still apply when the Miscellaneous teach pendant instruction LINEAR_MAX_SPEED is used.
- The robot will try to attain the maximum speed capability of at least one of its axes. It determines the maximum speed for the current move by comparing the teach pendant instruction LINEAR_MAX_SPEED with the maximum linear speed of 2000 mm/sec. The ratio of these two speeds is the percentage of the maximum axis speed that the axis will reach.

  For example: the maximum linear speed is 2000 mm/sec.

  ```
  1. LINEAR_MAX_SPEED = 1200
  2. L P[1] max_speed CNT1000
  ```

  The ratio of 1200 to 2000 is 60%. The system will drive the robot such that one of its axes will reach 60% of its maximum joint speed for line 2 of the program above.
- Max speed does not work with TCP speed prediction function (TCPP). That is, with Max speed, TCPP results may not be accurate.

15 SINGULARITY AVOIDANCE FUNCTION

For most of six-axis robots in FANUC, the singularity that occurs at elbow (at the boundary between "up" and "down" configurations) or at the boundary between "front" and "back" configurations causes regional structure degeneracy. This kind of singularity can easily be avoided by restricting the workplace of a robot. However, the wrist singularity may happen virtually any place inside the workplace. When a FANUC robot travels through/near a wrist singular position, motion performance becomes undesirable because

- Joint 4 and 6 will change a lot within a short time period;
- The TCP speed of the robot will slowdown;
- Path may deviate from the commanded one if a motor speed exceeds its limit.

Singularity avoidance function will provide a real-time solution to avoid the wrist singularity. With this function, the followings are achieved.

- The rotation of joint 4 and 6 is minimized, and robot can travel through/near a wrist singular position smoothly, and TCP speed can be maintained;
- This function works for both LINEAR TPE program motion and LINEAR jogging.

To use this function, singularity avoidance function option (A05B-2600-R792) or Motion package (A05B-2600-R809) is required. Some robot models do not support this function.

15.1 HOW TO USE SINGULARITY AVOIDANCE

How to use singularity avoidance in jogging and TPE program will be explained below.

15.1.1 How to Use Singularity Avoidance in Jogging

Enabled/disabled status of singularity avoidance function in jogging can be confirmed by checking whether “S/” is added before the manual-feed coordinate system shown in the teach pendant. Example: “S/WORLD”

If “S/” is added, the singularity avoidance in jogging is enabled. If “S/” is not added, the singularity avoidance in jogging is disabled. Enabled/disabled is automatically determined at power on depending on the selected program.

In order to select enabled/disabled, refer to the following procedure.

Procedure 15-1 Select enabled/disabled of singularity avoidance in jogging

Step

1 Press [FCTN] key, and select “T/Singularity JOG” to toggle enabled/disabled of singularity avoidance in jogging.

15.1.2 How to Use Singularity Avoidance in TPE Program

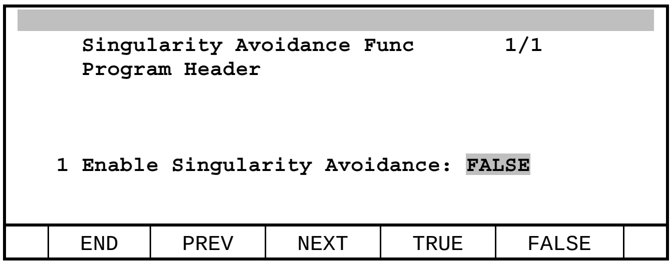
Enabled/disabled of singularity avoidance in TPE program can be selected for each program. When a new TPE program is created, the enabled/disabled of the program is automatically determined depending on the enabled/disabled status of singularity avoidance in jogging. Please note that in the case that the user loads a TPE program which is saved before the singularity avoidance option is added or is saved in the system which does not have the singularity avoidance option, the singularity avoidance is disabled for the TPE program.

In order to select enabled/disabled for each program, refer to the following procedure.

Procedure 15-2 Select enabled/disabled of singularity avoidance in program

Step

1. In the program select screen, move the cursor to the program and press the F2, DETAIL key to enter the detail screen.
2. Press the F3, NEXT key, and the following screen will be displayed. (This screen shows the singularity avoidance is disabled.)



```
   Singularity Avoidance Func       1/1
   Program Header


 1 Enable Singularity Avoidance: FALSE
```

3. In order to enable the singularity avoidance, press F4, TRUE. In order to disable the singularity avoidance, press F5, FALSE.

15.2 LIMITATIONS

Singularity avoidance function has the following limitations.

- This function is supported only in Handling Tool.
- Some robot models does not support this function.
- This function cannot be ordered with Coordinated motion option.
- This function cannot be used with Line tracking option.
- This function cannot be ordered with Continuous turn option.
- This function cannot be ordered with Shape GenerationII option.
- This function can be used only for linear motion. This function cannot be used for Circular/Arc motion.

15.3 CAUTIONS

Keep the following important information in mind when you set up and use singularity avoidance.

- With the singularity avoidance function, actual wrist configuration (flip/nonflip) might be different from the taught destination positions. The function might change configuration internally not only for the destination position in the motion line where singularity is detected but also for the subsequent destination positions in the following motions in the TP program.
- Since the function will change configuration internally, single step forward and backward might produce different behavior. To prevent the above from happening, during single step forward motion, the system will post a warning message "MOTN-208 Config Not Reached" at the point where the actual configuration is different from the taught one. By observing the message, the user should re-touch up the taught point at the specific line shown in the warning message this will update the taught configuration to the actual configuration. As a result, when stepping backward, the motion will behave the same way as stepping forward.
- Jogging and program motion might be different when the robot moves near singularity.
- If the destination position is inside singularity zone, the taught position is changed.
- The function might not help for the singularity at the corner path.

16 PATH SWITCHING FUNCTION

Path switching function (A05B-2600-J693) is the function to execute the assignment statements at the several points in a motion path.
When Path Switching (hereinafter called PS) instruction is specified in the DB calling program as follows, PS statements define the trigger point and the assignment statement to be executed.

DB CALL statement

```
  1: L P[1] 200mm/sec FINE DB 100mm, CALL A
```

Contents of sub-program A

```
  1:    PS -100mm +0.2sec,DO[1]=(ON)
  2:    PS -150mm -0.2sec,IF(DI[1]),DO[2]=(DI[2] AND !DI[3])
```

The distance specified just after DB statement does not affect to the each trigger timing of the PS commands in the sub-program. The PS statements in the sub program start their execution when the motion starts. User can mix another normal logical statement with the PS statement in the sub program.

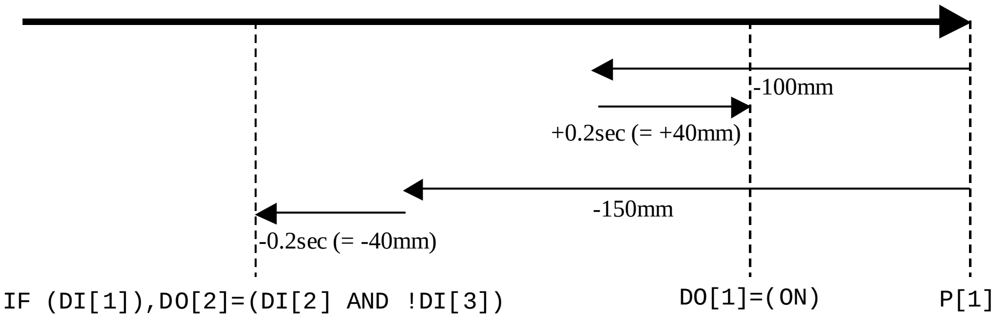


Fig. 16 (a) Path switching application

PS instruction is composed of two parts, trigger point and assignment statement.

```
PS -100mm +0.2sec,DO[1]=(ON)
```

Trigger point Assignment statement

Trigger point part specifies the distance (mm) from the destination position and the offset time (sec).
The negative value means the before the motion is completed.
The positive value means the after the motion is completed.
The offset time is converted to the distance according to the specified speed, in the above case, 0.2sec is converted to 40mm (= 200mm/sec * 0.2sec). This conversion is not precise in the acceleration and deceleration period.

According to the sign of the trigger distance value, its behavior is treated as distance-before or distance-after type. As described in the following section, trigger timing of distance-before is much different from that of distance-after type. And the time value to tune the trigger timing is affected by override value. So to avoid unexpected trigger by override change, if the initial distance value is minus value, regardless of the time value, actual PS trigger point will be clamped to -0.0mm as distance-before type, internally. And if the initial distance value is plus value, regardless of the time value, actual PS trigger point will be clamped to +0.0mm as distance after type.
Assignment statement part specifies the statement to be executed at the trigger point.
The assignment statement can have IF condition.

```
PS -150mm -0.2sec,IF (DI[1]),DO[1]=(ON)
```

In this case, when DI[1] is ON, DO[1] is turned on, when DI[1] is OFF, DO[1] is not changed.
The assignment statement can use the following items and operators.

Items for left side	Operator for right side or IF condition	Items for right side or IF condition
DO[]	( )	ON OFF
RO[]	+	Constant value
WO[]	-	DI[] DO[]
AO[]	*	RI[] RO[]
GO[]	/	WI[] WO[]
SO[]	MOD	AI[] AO[]
UO[]	DIV	GI[] GO[]
F[]	AND	SI[] SO[]
R[]	OR	UI[] UO[]
PR[i,j]	!	F[]
System variables	= (comparison)	M[]
M[]	<	R[]
TC_ONLINE	>	PR[i,j]
	<=	TIMER[]
	>=	System variables
	<>	TCP_SPD[]

Marker assignment and TC_ONLINE can be specified in PS statement.
AR[] (calling parameter) can not be specified in PS statement.

Up to 20 operations can be specified in one motion line as total of right side formula and IF condition.
One motion statement can have up to 20 PS statements.

Case that the trigger point is negative and it is farther than the start position (distance-before type)

If the trigger point is negative and it is farther than the start position, this is triggered immediately.

```
1: L P[1] 200mm/sec FINE
2: L P[2] 200mm/sec FINE DB 100mm, CALL A
```

Contents of sub-program A

```
1:   PS -1000mm +0sec,DO[1]=(ON)
```

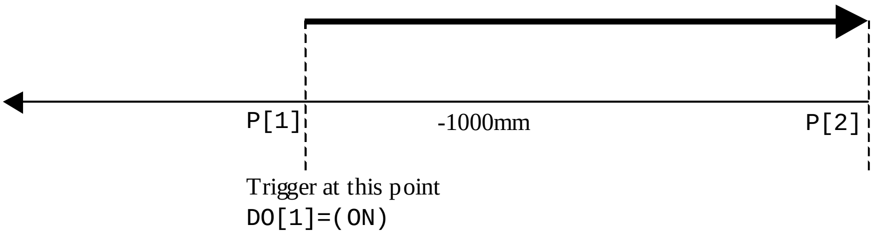


Fig. 16 (b) Path switch application

You can choose to cause error at this situation by following parameter setting.

$PS_CONFIG.$DB_IMMTRIG (distance-before type)
$PS_CONFIG.$DA_IMMTRIG (distance-after type)

Value	Description
0	Trigger PS condition immediately, but post neither alarm nor warning.
1 (default)	Trigger PS condition immediately, and post “INTP-539 PS(program name, line number, distance[mm]) Already in area” (warning).
other than (0, 1)	Not triggered, and post “INTP-540 PS(program name, line number, distance[mm]) Already in area” (PAUSE).

Case that the trigger point is passed by jog while the program is paused

When original path resume is disabled ($SCR.$ORG_PTH_RSM=FALSE), if the program is paused during the motion with PS, and jogged by TP, and the trigger point is passed by jog, then the PS condition is triggered immediately when the program is resumed.

When original path resume is enabled ($SCR.$ORG_PTH_RSM=TRUE), once back to the paused position on the original motion path and re-start the program. In this case, the PS condition is triggered when the trigger point is passed after resume and re-start.

You can choose to cause error at this situation by above parameter setting.
($PS_CONFIG. $DB_IMMTRIG, $DA_IMMTRIG)

```
    1: L P[1] 200mm/sec FINE
    2: L P[2] 200mm/sec FINE DB 100mm, CALL A
```

Contents of sub-program A

```
    1:    PS -500mm +0sec,DO[1]=(ON)
```

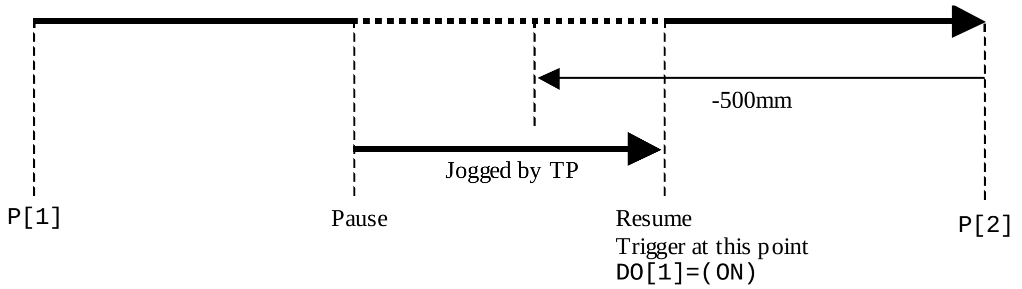


Fig. 16 (c) Path switch application (original path resume is disabled)

Case that TCP path moves away without entering the trigger area on CNT motion

In case of CNT motion, teaching point does not on the actual robot path. If the robot path is too far from the trigger area, reminding untriggered distance-before type PS conditions are triggered when the TCP path starts to move away from the teaching point.

```
    1: L P[1] 200mm/sec FINE
    2: L P[2] 200mm/sec CNT100 DB 100mm, CALL A
    3: L P[3] 200mm/sec FINE
```

Contents of sub-program A

```
    1:   PS -10mm +0sec,DO[1]=(ON) (distance-before type)
    2:   PS +10mm +0sec,DO[2]=(ON) (distance-after type)
```

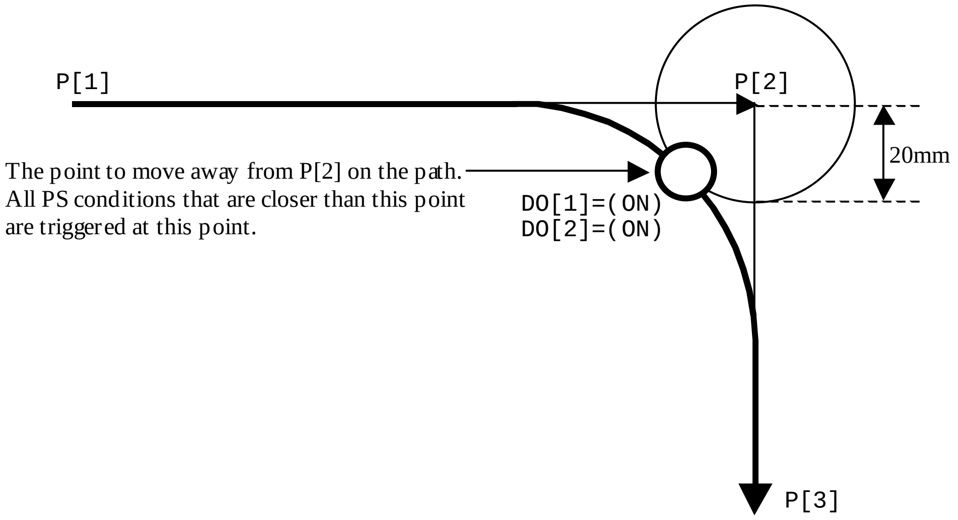

P[1]
P[2]
20mm
The point to move away from P[2] on the path.
All PS conditions that are closer than this point
are triggered at this point.
DO[1]=(ON)
DO[2]=(ON)
P[3]

Fig. 16 (d) Path switch application

You can choose to cause error at this situation by following parameter setting.

$PS_CONFIG.$DB_NOTRIG (distance-before type)
$PS_CONFIG.$DA_NOTRIG (distance-after type)

Value	Description
0	Trigger PS condition at the closest point, but post neither alarm nor warning.
1 (default)	Trigger PS condition at the closest point, and post “INTP-541 PS(program name, line number, distance[mm]) Forced trigger” (warning).
other than (0, 1)	Not triggered, and post “INTP-542 PS(program name, line number, distance[mm]) No trigger” (PAUSE).

Case that the trigger point is positive and it is farther than the destination position (distance-after type)

If the trigger point is positive and it is farther than the destination position of the next motion, the PS condition is triggered when the next motion is completed.

You can choose to cause error at this situation by above parameter setting.
($PS_CONFIG. $DB_NOTRIG, $DA_NOTRIG)

```
1: L P[1] 200mm/sec FINE
2: L P[2] 200mm/sec FINE DB 100mm, CALL A
3: L P[3] 200mm/sec FINE
```

Contents of sub-program A

```
 1:    PS +1500mm +0sec,DO[1]=(ON)
```

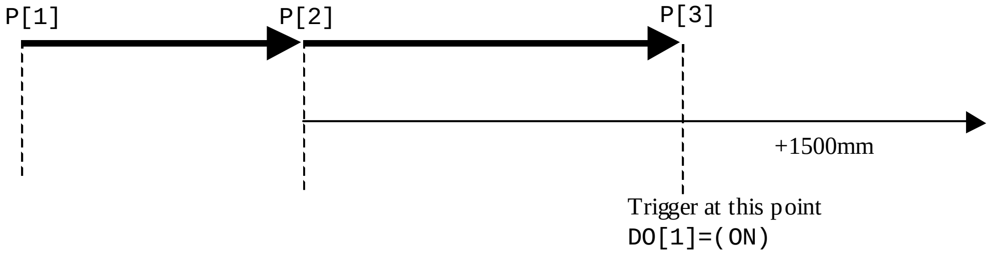


Fig. 16 (e) Path switch application

About multi group configuration

In case of multi group configuration, the PS statement is implicitly executed for the first motion group in the group mask of the TP program. For example, if the motion mask is [*,1,1,*,*,*,*,*], then the PS statement is executed for group 2.

Limitations

PS statement is not available for the following motion type.

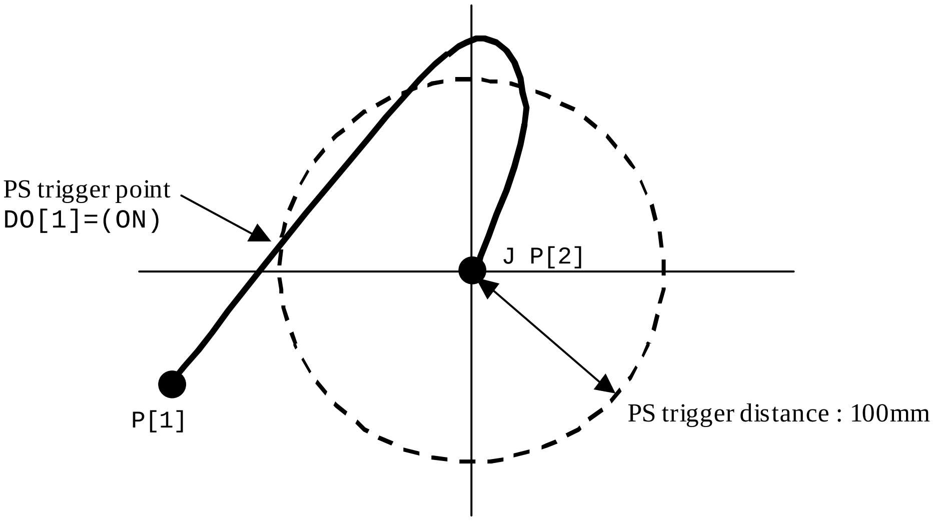
- Weaving motion
- Tracking motion which can not define actual destination point beforehand, as line-tracking, MIG-EYE, Robot-Link.

Caution

- PS statement is interpreted in a flash only at the start of the new motion line. When the motion including PS statement is paused, if you edit and change the PS statement in the sub program during this pause status, modified PS commands do not take effect until the motion line is newly executed. After pausing program, to take effect of the modified PS statement immediately, you can BWD the motion line and re-start. Then the modified PS statements will take effect.

- PS command for joint motion is not prohibited. But it must be used carefully. Regardless of the motion type Linear, Circular, or Joint, PS command is checking the linear distance between TCP and destination point as trigger threshold.

> ⚠ **NOTE**
> In cases of PS command's Distance-Before type usage, If TCP once goes into the defined sphere around the destination point, the PS command will be triggered regardless how long the rest of path. Please note the distance which is being checked in PS motion is not the total remaining path length to the destination point. Please see the following figure for example.



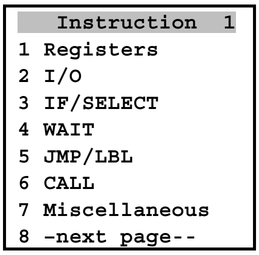
```
1: J P[1] 100% FINE
2: J P[2] 100% FINE DB 100mm, CALL PROG1
```

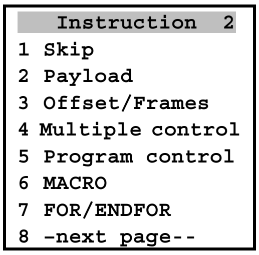
Contents of sub-program PROG1

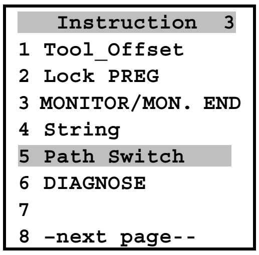
```
1: PS -100mm +0.0sec, DO[1] = ON
```

TP editor operation

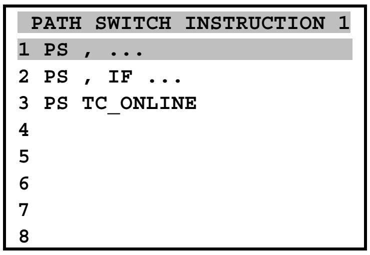
1 "Path Switching" item is displayed in the [INST] menu.



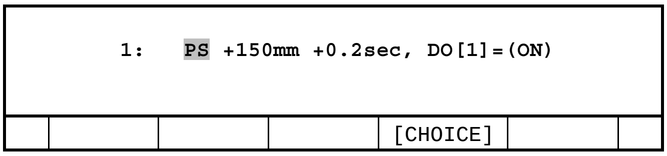




2 Select "Path Switching" item, the following menu will be displayed.

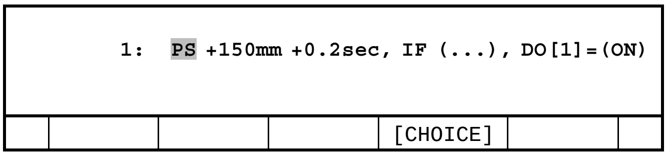


3 The distance and offset time can be set, and the assignment statement can be edited as the same way as mixed logic instructions.



```
1:   PS +150mm +0.2sec, DO[1]=(ON)
```

4 When the cursor is on 'PS ' item, you can switch "PS ..." to "PS ... IF ..." and vice versa from menu.



```
1:  PS +150mm +0.2sec, IF (...), DO[1]=(ON)
```

17 AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION

In the robot system that auxiliary axis controls a jig, there is a requirement to cut off the power of auxiliary axis for safety when an operator does a job with that jig. Auxiliary axis servo off (Local stop) is a function for that purpose. This function can turn off motor power supply for each auxiliary axis respectively and prevent axis from unexpected movements.

This function has following merits.

- Keep safety for operator who works at jig (auxiliary axis).
- Improve the cycle time of process because it is not necessary to stop the operation of the robot when the operator enters in Load station to replace the work piece.
- Output servo ON/OFF signal by just using macro program or press button.

To use this function, auxiliary axis servo off option (A05B-2600-J806) and dedicated hardware and setup at DCS Local stop menu are required. In regard to the dedicated hardware or DCS Local stop menu, please refer to the following manuals.

- R-30*i*B Local Stop function with STO Maintenance and Order Manual (A-95028E)
- Chapter “AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION” in R-30*i*B/R-30*i*B Mate CONTROLLER Dual Check Safety Function OPERATOR’S MANUAL (B-83184EN)

17.1 SPECIFICATION

- Order number for software option
  A05B-2600-J806

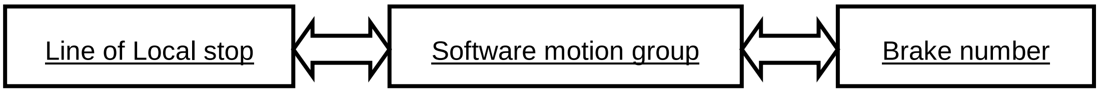
- Way to cut off motor power supply
  This function cut off power from servo amplifier to motor. Even in the Local stop mode (power supply for motor is cut off), power is supplied to the amplifier.

- Motor for the auxiliary axis
  Please be sure to use motor with brake for this function and set brake number independent of other motion group when you set up auxiliary axis.

- Local stop Hardware
  A dedicated hardware is required. In regard to the dedicated hardware, please refer to R-30*i*B Local Stop function with STO Maintenance and Order Manual (A-95028E).

- Setup DCS Local stop menu
  To use this function, please be sure to setup at DCS Local stop menu. In regard to DCS Local stop menu, please refer to R-30*i*B Local Stop function with STO Maintenance and Order Manual (A-95028E) and chapter “AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION” in R-30*i*B/R-30*i*B Mate CONTROLLER Dual Check Safety Function OPERATOR’S MANUAL (B-83184EN).

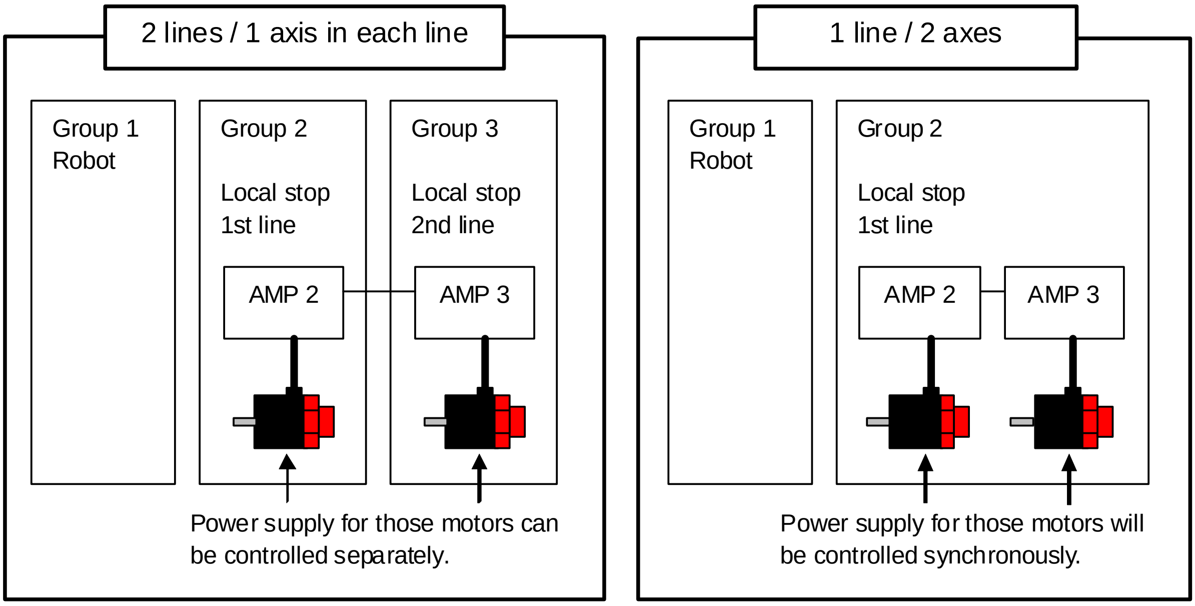
- Line of Local stop
  The unit (group) in which power supply of motors is controlled synchronously is called as Line of Local stop. Line concept corresponds to the software motion group. Motors in the same line must have the same brake number and motors in the different line must have the different brake number. Software supports 8 Local stop lines at maximum.



- Number of axes in the Local stop line
  If there are multiple axes in the one Local stop line, power supply for those motors will be controlled synchronously. In this case, there are multiple axes in one motion group. Power supply for those motors cannot be controlled separately. If you want to control separately, please separate a Local stop line.

> Configuration of Local stop function is determined by the number of lines and the number of axes in one line.

- Example of system configuration
  When there are two axes that use Local stop function, possible configurations are as follows.


2 lines / 1 axis in each line
Group 1
Robot
Group 2
Local stop
1st line
AMP 2
Group 3
Local stop
2nd line
AMP 3
Power supply for those motors can be controlled separately.
1 line / 2 axes
Group 1
Robot
Group 2
Local stop
1st line
AMP 2
AMP 3
Power supply for those motors will be controlled synchronously.

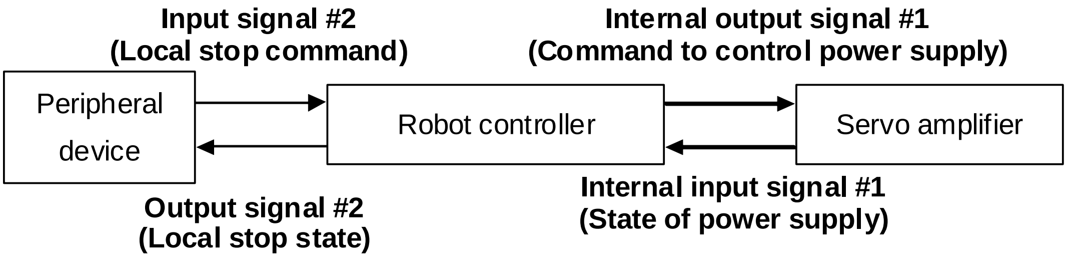
- Necessary signals for Local stop function
  One input signal and one output signal are required for each Local stop line.

  1 Input signal #2
  Local stop command signal by the user.
  Local stop software controls power supply for the motor in response to this signal.
  - When this signal is ON, supply power to the motor and the axis can move.
  - When this signal is OFF, cut off power to the motor and the axis cannot move.

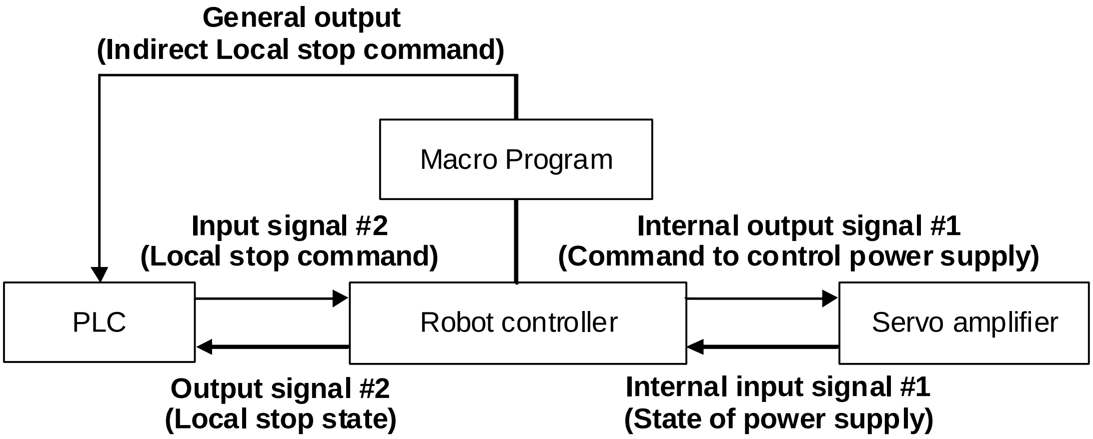
  2 Output signal #2
  Local stop status feedback signal from a robot controller to a peripheral device such as PLC.
  This signal shows the state of Local stop software controls power supply for the motor. Please use this signal as muting indicator and WAIT condition of wait for transit servo off/on to complete in TP program. Please do not operate this signal.
  - When this signal is OFF, power is supplied to the motor and the axis can move.
  - When this signal is ON, power is not supplied to the motor and the axis cannot move.

  Please assign general I/O or robot I/O for input signal #2 and output signal #2.

- Basic flow of process
  When the user inputs the input signal #2, software outputs the internal output signal #1 to the amplifier and power supply for motor will be controlled. Software also outputs the output signal #2 to a peripheral device after checking the state of power supply in amplifier by the internal input signal #1. Please connect a peripheral device such as push button to the input signal #2.



In case a macro program will input that Local stop command, please output general output signal in that program to the peripheral device such as PLC and control input signal #2 according to that output.



17.2 CONSTRAINTS

- “LSTP-001 Motion grp %d is active” alarm occurs when the user command cutting off the power supply (input signal #2 = OFF) while auxiliary axis is moving. Then running program will be paused and robot stops when this alarm occurs. Please command power off while axis is not moving.
- “LSTP-011 Motion grp %d is in LSTOP” alarm occurs when the user tried to move axis while the power supply is cut off. Then axis does not move. Please turn on power supply for auxiliary axis before you move axis.
- Please be sure to specify FINE positioning path for last motion instruction right before command power off (input signal #2 = OFF). If specify CNT positioning path for last motion instruction right before command power off, “LSTP-001 Motion grp %d is active” alarm might occur.
- Please be sure to keep time more than 0.2 sec before command power off (input signal #2 = OFF). If command power off right after last motion instruction without waiting 0.2 sec, “LSTP-001 Motion grp %d is active” alarm might occur.

17.3 SETTINGS

1 Firstly, please connect a dedicated hardware. In regard to the dedicated hardware, please refer to R-30*i*B Local Stop function with STO Maintenance and Order Manual (A-95028E).

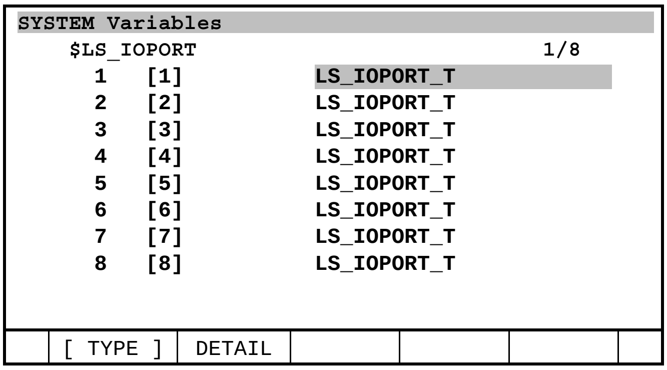
2 Secondly, please setup DCS Local stop menu. In regard to DCS Local stop menu, please refer to R-30*i*B Local Stop function with STO Maintenance and Order Manual (A-95028E) and chapter "AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION" in R-30*i*B/R-30*i*B Mate CONTROLLER Dual Check Safety Function OPERATOR'S MANUAL (B-83184EN).

3 Thirdly, please set system variables. For each Local stop line, setting of motion group number, input signal #2, output signal #2 and internal output signal #1 are necessary. For input signal #2 and output signal #2, please specify port type and port number. For internal output signal #1, please specify the SPO index number. In regard to the SPO index number, please refer to R-30*i*B Local Stop function with STO Maintenance and Order Manual (A-95028E) and chapter "AUXILIARY AXIS SERVO OFF (LOCAL STOP) FUNCTION" in R-30*i*B/R-30*i*B Mate CONTROLLER Dual Check Safety Function OPERATOR'S MANUAL (B-83184EN).

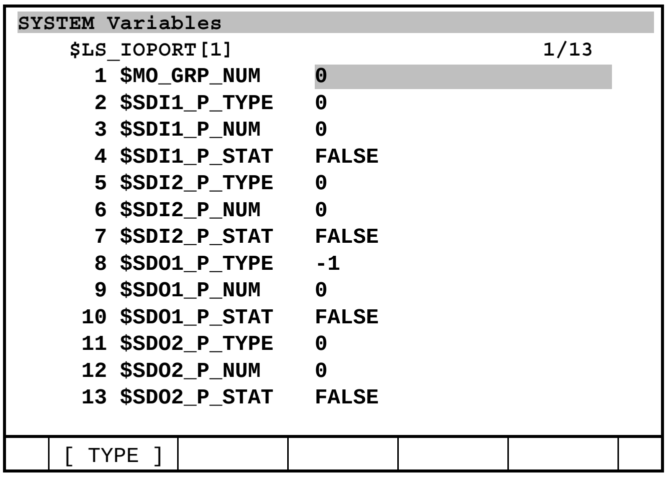
Procedure to set up:

1 Display the SYSTEM Variables screen.
a) Press [MENU] key.
b) Select 0 NEXT and then 6 SYSTEM.
c) Press F1, [TYPE] key.
d) Select Variables, and then SYSTEM variables screen will be shown.

2 Move the cursor to $LS_IOPORT and press [ENTER] key. The following screen will be displayed.


```
SYSTEM Variables
    $LS_IOPORT                            1/8
      1   [1]          LS_IOPORT_T
      2   [2]          LS_IOPORT_T
      3   [3]          LS_IOPORT_T
      4   [4]          LS_IOPORT_T
      5   [5]          LS_IOPORT_T
      6   [6]          LS_IOPORT_T
      7   [7]          LS_IOPORT_T
      8   [8]          LS_IOPORT_T
```

3 Select the Local stop line (1-8) that needs to be configured and press [ENTER] key. The following screen will be displayed.


```
SYSTEM Variables
    $LS_IOPORT[1]                         1/13
      1 $MO_GRP_NUM     0
      2 $SDI1_P_TYPE    0
      3 $SDI1_P_NUM     0
      4 $SDI1_P_STAT    FALSE
      5 $SDI2_P_TYPE    0
      6 $SDI2_P_NUM     0
      7 $SDI2_P_STAT    FALSE
      8 $SDO1_P_TYPE    -1
      9 $SDO1_P_NUM     0
     10 $SDO1_P_STAT    FALSE
     11 $SDO2_P_TYPE    0
     12 $SDO2_P_NUM     0
     13 $SDO2_P_STAT    FALSE
```

4 Please refer to the below tables and setup the system variables according to the actual I/O connection. It is required to setup system variables which "O" (Need to setup) is written in rightmost line in the following Table 17.3. For system variables which "X" (No need to setup) is written, please do not change.

Table 17.3 Description of system variable (Following settings are necessary for each Local stop line)

Variable name	Description	O: Need to setup X: No need to setup
$MO_GRP_NUM	Motion group number that uses a Local stop function. Valid range is 2-8. Setting this variable to 0 disables Local stop function for the Local stop line.	O
$SDI1_P_TYPE	Not in use.	X
$SDI1_P_NUM	Not in use.	X
$SDI1_P_STAT	Status of internal input signal #1. Please do not change this variable.	X
$SDI2_P_TYPE	Port type for input signal #2 (Local stop command signal by a user).	O
$SDI2_P_NUM	Port number for input signal #2.	O
$SDI2_P_STAT	Status of input signal #2. Please do not change this variable.	X
$SDO1_P_TYPE	Signal type for internal output signal #1 (Command to control power supply from a robot controller to servo amplifier). Please do not change from the default value (-1).	X
$SDO1_P_NUM	Specify the SPO index number. In regard to the SPO index number, please refer R-30*i*B Local Stop function with STO Maintenance and Order Manual (A-95028E).	O
$SDO1_P_STAT	Status of internal output signal #1. Please do not change this variable.	X
$SDO2_P_TYPE	Port type for output signal #2 (Local stop status feedback signal from a robot controller).	O
$SDO2_P_NUM	Port number for output signal #2.	O
$SDO2_P_STAT	Status of output signal #2. Please do not change this variable.	X

Values for port type ($LS_IOPORT[n].$SDI2_P_TYPE, $SDO2_P_TYPE) are as follows. Please assign general I/O or robot I/O for the input signal #2 and output signal #2.

Value for port type	Type of signal
1	DI (General input signal)
2	DO (General output signal)
8	RI (Robot input signal)
9	RO (Robot output signal)

Example of configuration
In case that assign DI[10] for the input signal #2 and DO[11] for the output signal #2 for Local stop line 1, set the above system variables as follows.
$LS_IOPORT[1].$SDI2_P_TYPE = 1
$LS_IOPORT[1].$SDI2_P_NUM = 10
$LS_IOPORT[1].$SDO2_P_TYPE = 2
$LS_IOPORT[1].$SDO2_P_NUM = 11

5 When you use more than 2 Local stop lines, please set system variables for 2nd line or later too.
6 Please do not change $LS_CONFIG and $LS_SYSTEM.
7 That is all for software setup. In order to make settings available, please cycle power once. After cycling power, setting will be available.
8 To confirm settings are proper, please do check by following procedure for each Local stop lines.

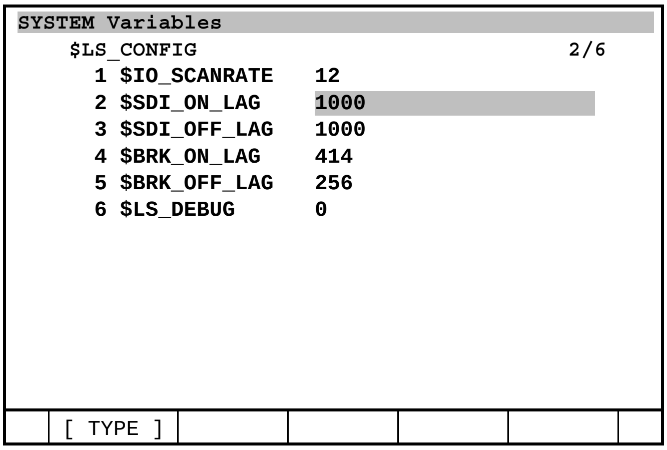
Confirmation 1) Exiting from Local stop mode
When you set the input signal #2 to ON, the output signal #2 becomes OFF.

Confirmation 2) Entering into Local stop mode
When you set the input signal #2 to OFF, the output signal #2 becomes ON.

> ⚠**NOTE**
> Please perform these checks when the system is NOT in servo off status.
> If any alarm factor such as Operator panel E-stop, Teach pendant E-stop, deadman switch release, External emergency stop, etc. exists, please clear these alarms first. Then perform a test when the system is in servo on status.

9 If no alarm occurs and signals change like above, Local stop function works correctly.

10 Following alarms could occur.
"LSTP-006 DI1 ON timer expired (G:%d)" or "LSTP-007 DI1 OFF timer expired (G:%d)"
In case that
"LSTP-006 DI1 ON timer expired (G:%d)" occurs but the input signal #1 becomes ON or
"LSTP-007 DI1 OFF timer expired (G:%d)" occurs but the input signal #1 becomes OFF,
please increase following system variables
$LS_CONFIG.$SDI_ON_LAG and $LS_CONFIG.$SDI_OFF_LAG 100 by 100 until alarm does not occur. The default value for these variables is 1000. If you change the value, please set the same value to both variables.



```
SYSTEM Variables
    $LS_CONFIG                                  2/6
      1 $IO_SCANRATE   12
      2 $SDI_ON_LAG    1000
      3 $SDI_OFF_LAG   1000
      4 $BRK_ON_LAG    414
      5 $BRK_OFF_LAG   256
      6 $LS_DEBUG      0
```

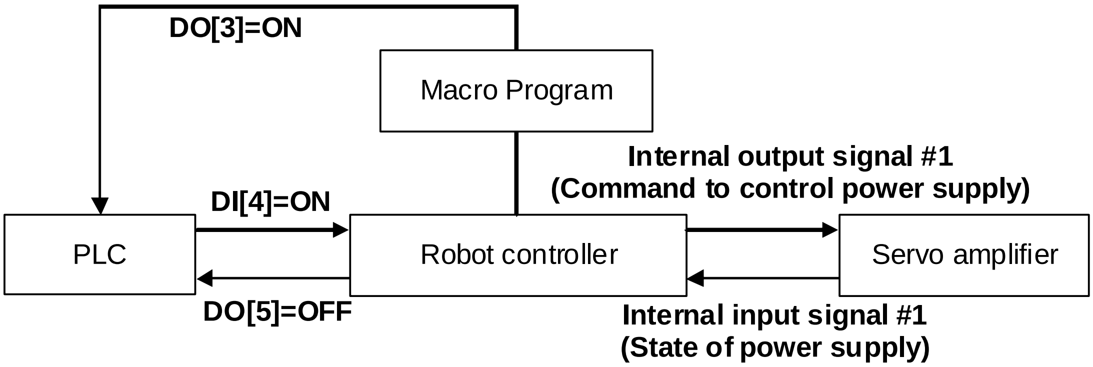
When "LSTP-006 DI1 ON timer expired (G:%d)" or "LSTP-007 DI1 OFF timer expired (G:%d)" occurs, it is necessary to cycle power once to clear alarm status.

11 If alarms occur except for the above situation, please refer to FANUC Robot series R-30*i*B/R-30*i*B Mate CONTROLLER OPERATOR'S MANUAL (Alarm Code List) (B-83284EN-1) and check the settings of system variable, setting of brake number, and hardware connections.

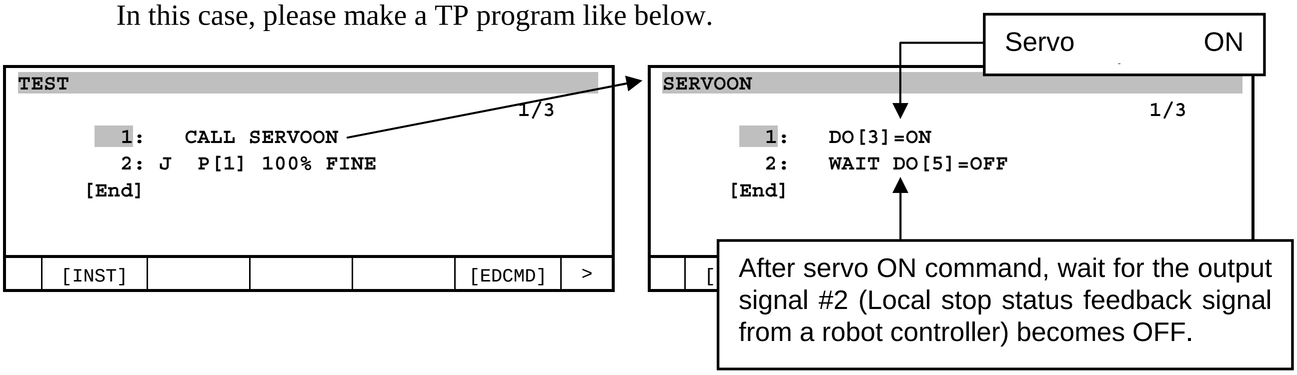
17.4 ATTENTION

- Note for moving auxiliary axis right after exiting from Local stop mode (servo ON command)
  Please do not move auxiliary axis until output signal #2 becomes OFF.

  Example) Assuming the system in which I/O has been set up like below.



In this case, please make a TP program like below.



```
TEST
                                        1/3
     1:   CALL SERVOON
     2: J  P[1] 100% FINE
    [End]
```

```
SERVOON
                                        1/3
     1:   DO[3]=ON
     2:   WAIT DO[5]=OFF
    [End]
```

Servo ON

After servo ON command, wait for the output signal #2 (Local stop status feedback signal from a robot controller) becomes OFF.

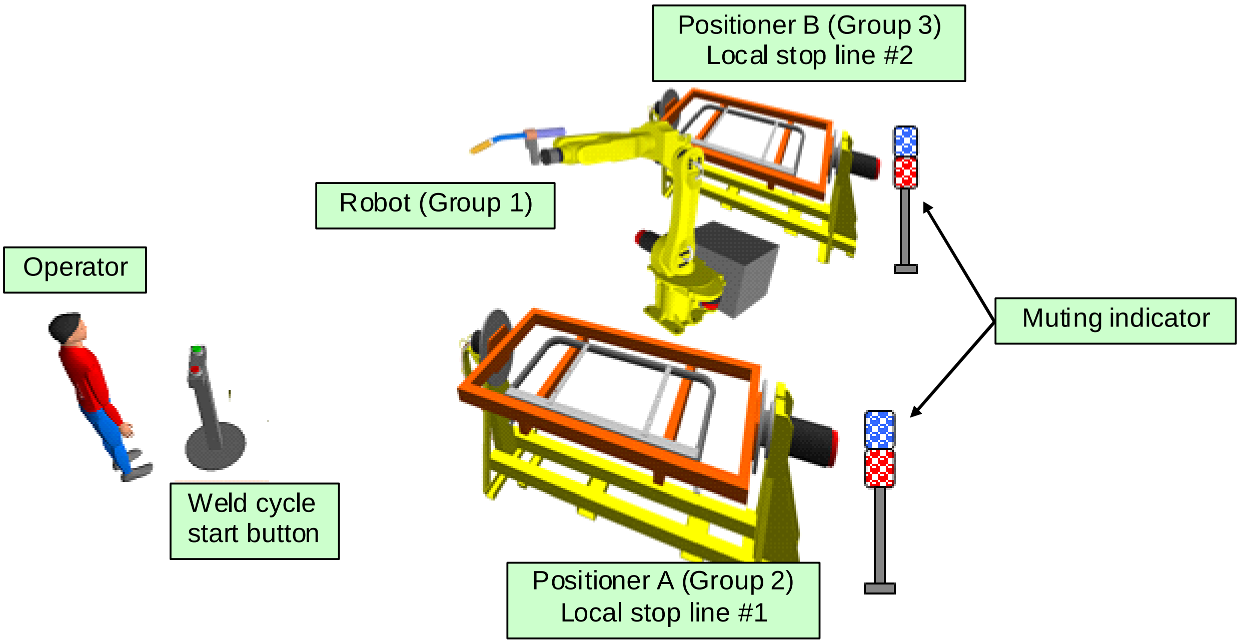
If you teach motion instruction right after servo ON command without WAIT instruction, “LSTP-011 Motion grp %d is in LSTOP” alarm might occur.

- Motion group mask of a TP program in which servo OFF will be commanded.
  Please set a motion group mask to (*,*,*,*,*,*,*,*) for a TP program in which servo OFF will be commanded to a certain auxiliary axis group by RO or DO. (NOTE: Please set a motion group mask at creating a program.)
- When turning on a robot controller, servo ON command will be output once.
  This is due to the initialization of software internal status. If the input signal #2 is ON, the status will be kept (servo ON). If the input signal #2 is OFF, servo OFF command will be output and auxiliary axis will enter into the Local stop mode (servo OFF).

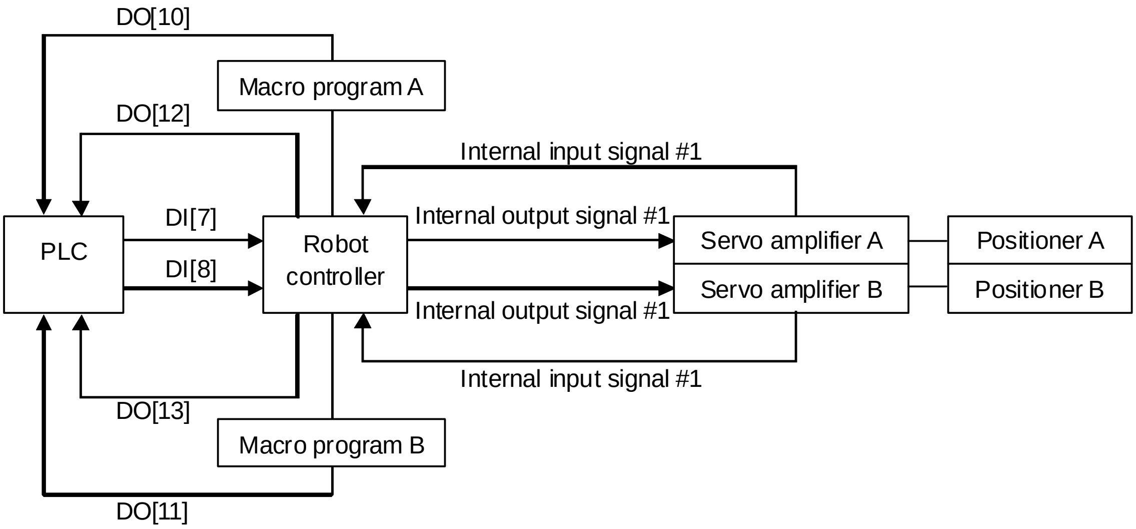
17.5 PROGRAMMING

Suppose a system like below.

- 3 groups welding system that consists of a robot and two positioners (A and B).
- Operator loads/unloads a workpiece at one side of the positioner while robot welds at another positioner.
- 2 Local stop lines are required.



- Assume that I/O is connected like below.



Suppose a welding sequence as follows.

A) Robot completes welding of a workpiece B at positioner B (Group 3) and returns to a home position.

B) An operator confirms that workpiece A of positioner A (Group 2) is ready and starts program A by pressing cycle start button.

C) WELD_A.TP program starts.

C.1 In macro UNLOCK_A, positioner A exits from Local stop mode.

C.2 In LOAD_B.TP, positioner B moves to the position for loading/unloading workpiece B.

C.3 In macro LOCK_B, positioner B enters into Local stop mode.

C.4 In WELD_G2.TP, robot starts to weld a workpiece A at positioner A.

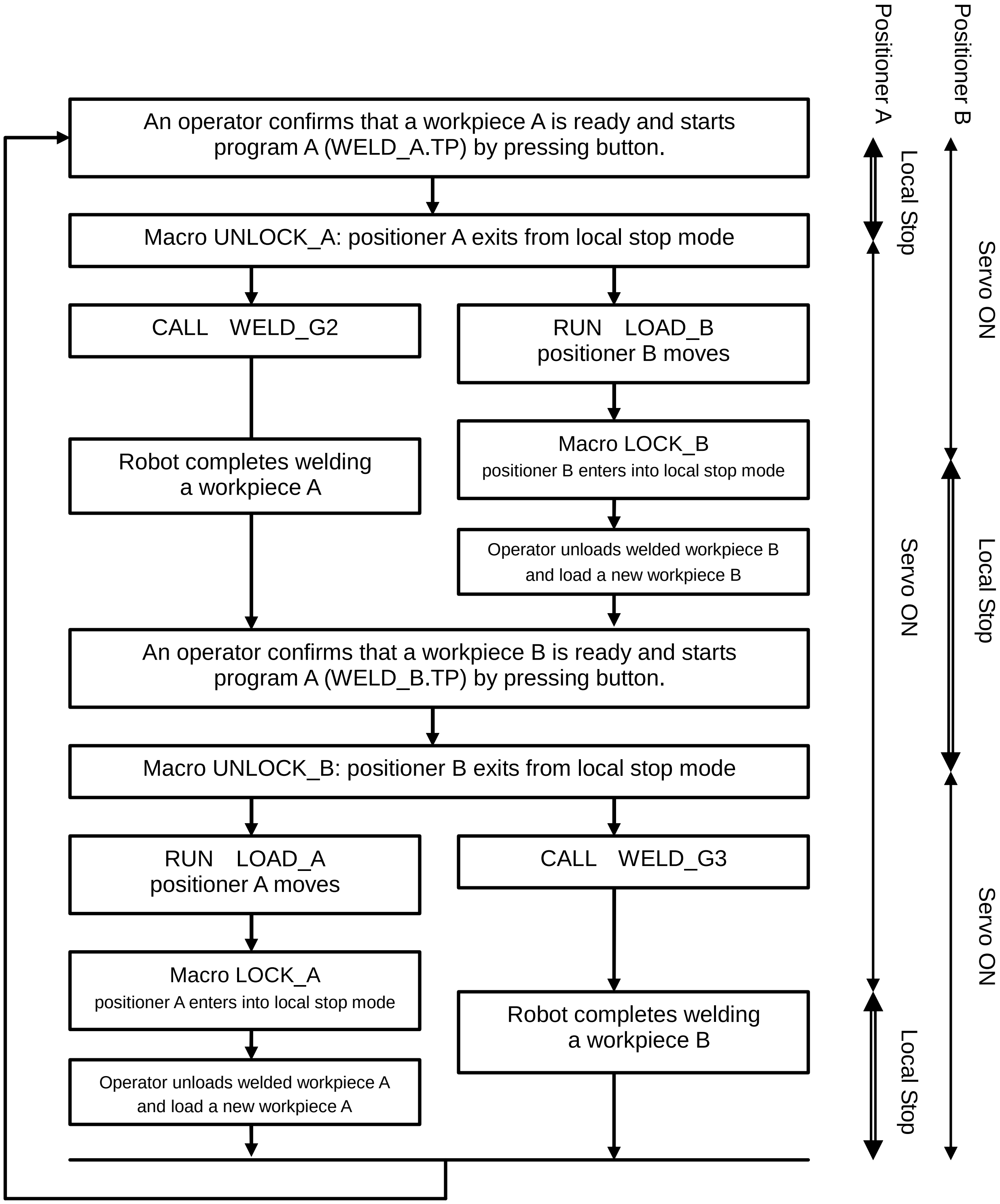
C.5 An operator confirms that a muting indicator (This indicator is connected to the output signal #2) indicates positioner B is in Local stop mode. Then an operator unloads welded workpiece B and loads a new workpiece B.

D) Robot completes welding of a workpiece A at positioner A (Group 2) and returns to a home position.

E) An operator confirms that a workpiece B is ready and starts next program B by pressing button.

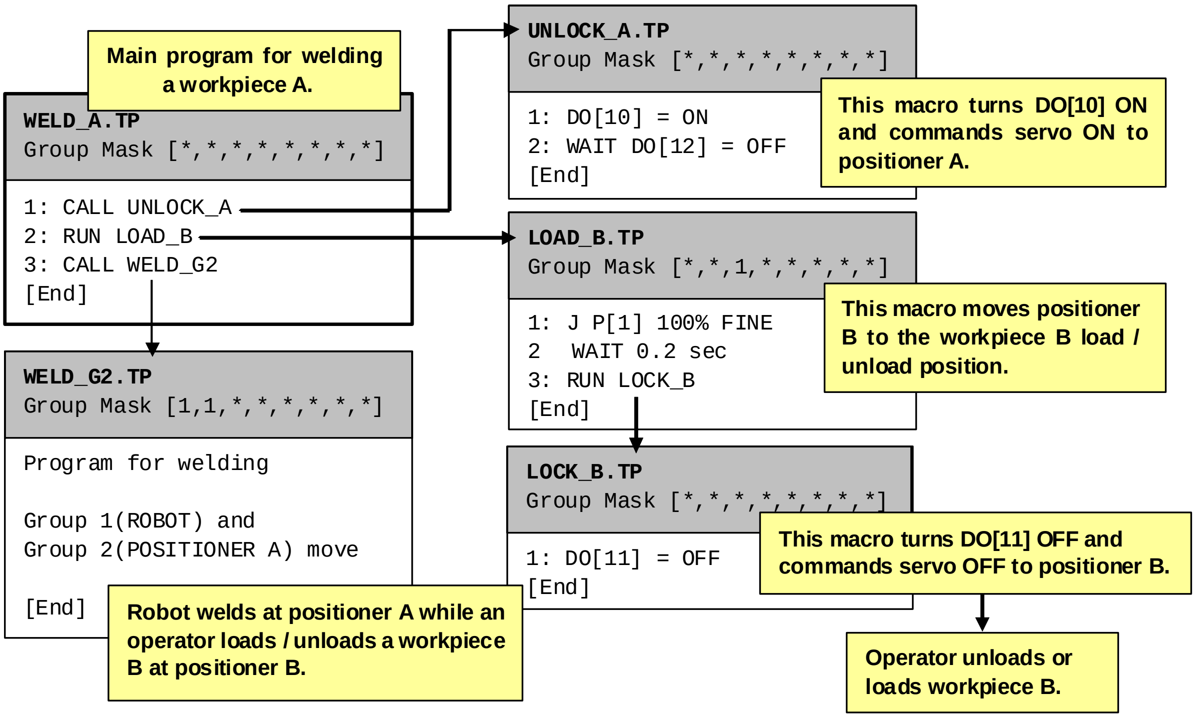
F) Repeat C)~D) and then go back to program A.

Cycle flow of the above-mentioned example.


An operator confirms that a workpiece A is ready and starts program A (WELD_A.TP) by pressing button.
Macro UNLOCK_A: positioner A exits from local stop mode
CALL WELD_G2
RUN LOAD_B positioner B moves
Macro LOCK_B positioner B enters into local stop mode
Robot completes welding a workpiece A
Operator unloads welded workpiece B and load a new workpiece B
An operator confirms that a workpiece B is ready and starts program A (WELD_B.TP) by pressing button.
Macro UNLOCK_B: positioner B exits from local stop mode
RUN LOAD_A positioner A moves
CALL WELD_G3
Macro LOCK_A positioner A enters into local stop mode
Robot completes welding a workpiece B
Operator unloads welded workpiece A and load a new workpiece A
Positioner A
Positioner B
Local Stop
Servo ON
Servo ON
Local Stop
Servo ON
Local Stop

For example, following programs accomplish this sequence.

Program A for welding of a workpiece A : WELD_A



Main program for welding a workpiece A.

```
WELD_A.TP
Group Mask [*,*,*,*,*,*,*,*]

1: CALL UNLOCK_A
2: RUN LOAD_B
3: CALL WELD_G2
[End]
```

```
WELD_G2.TP
Group Mask [1,1,*,*,*,*,*,*]

Program for welding

Group 1(ROBOT) and
Group 2(POSITIONER A) move

[End]
```

Robot welds at positioner A while an operator loads / unloads a workpiece B at positioner B.

```
UNLOCK_A.TP
Group Mask [*,*,*,*,*,*,*,*]

1: DO[10] = ON
2: WAIT DO[12] = OFF
[End]
```

This macro turns DO[10] ON and commands servo ON to positioner A.

```
LOAD_B.TP
Group Mask [*,*,1,*,*,*,*,*]

1: J P[1] 100% FINE
2  WAIT 0.2 sec
3: RUN LOCK_B
[End]
```

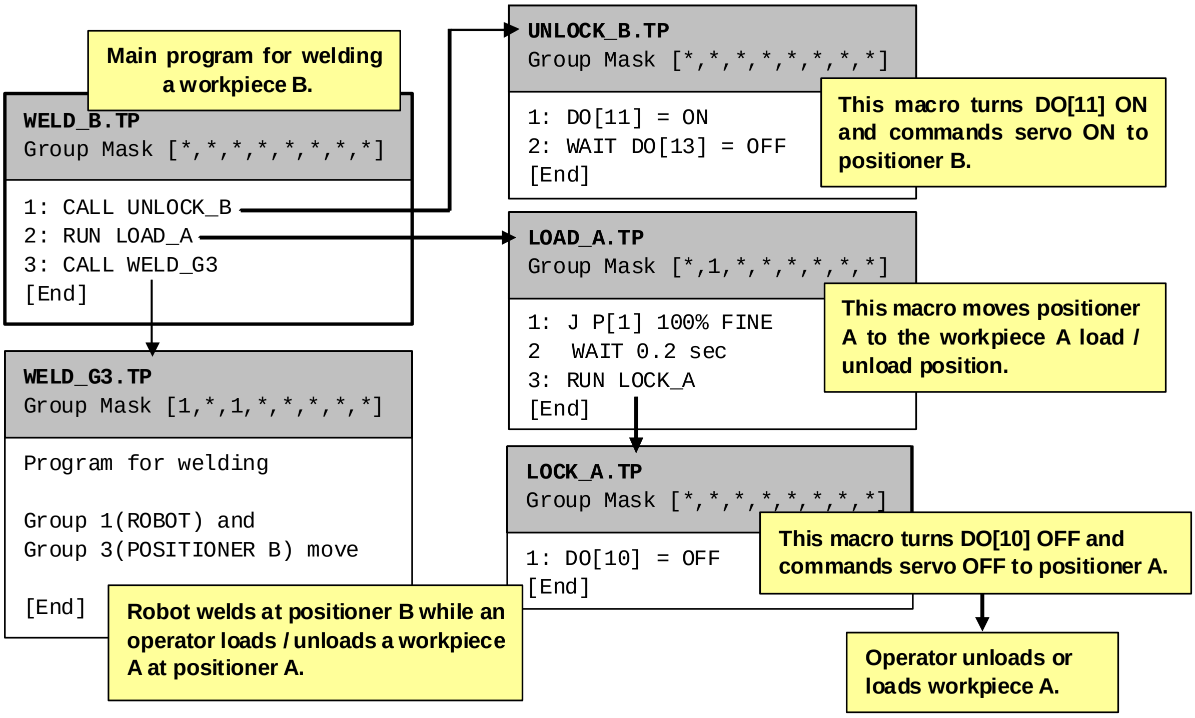
This macro moves positioner B to the workpiece B load / unload position.

```
LOCK_B.TP
Group Mask [*,*,*,*,*,*,*,*]

1: DO[11] = OFF
[End]
```

This macro turns DO[11] OFF and commands servo OFF to positioner B.

Operator unloads or loads workpiece B.

Program B for welding of a workpiece B : WELD_B



Main program for welding a workpiece B.

```
WELD_B.TP
Group Mask [*,*,*,*,*,*,*,*]

1: CALL UNLOCK_B
2: RUN LOAD_A
3: CALL WELD_G3
[End]
```

```
WELD_G3.TP
Group Mask [1,*,1,*,*,*,*,*]

Program for welding

Group 1(ROBOT) and
Group 3(POSITIONER B) move

[End]
```

Robot welds at positioner B while an operator loads / unloads a workpiece A at positioner A.

```
UNLOCK_B.TP
Group Mask [*,*,*,*,*,*,*,*]

1: DO[11] = ON
2: WAIT DO[13] = OFF
[End]
```

This macro turns DO[11] ON and commands servo ON to positioner B.

```
LOAD_A.TP
Group Mask [*,1,*,*,*,*,*,*]

1: J P[1] 100% FINE
2  WAIT 0.2 sec
3: RUN LOCK_A
[End]
```

This macro moves positioner A to the workpiece A load / unload position.

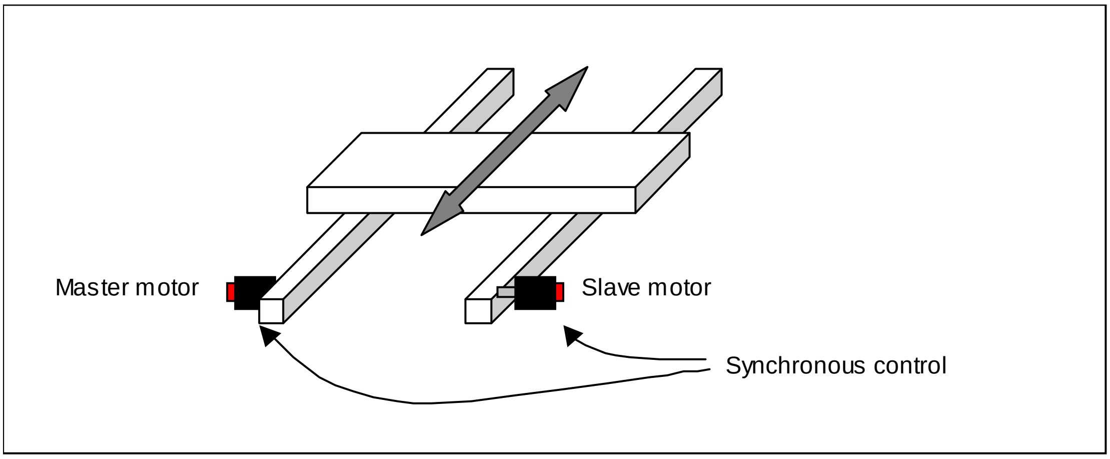
```
LOCK_A.TP
Group Mask [*,*,*,*,*,*,*,*]

1: DO[10] = OFF
[End]
```

This macro turns DO[10] OFF and commands servo OFF to positioner A.

Operator unloads or loads workpiece A.

18 DUAL MOTOR DRIVE

The Dual Motor Drive (Dual Drive) feature allows single axis control of two motors. One motor is called the master motor and the other motor is called the slave motor.



- The main function of the Dual Drive feature is to compensate for synchronous error due to servo delay between master and slave axes.
- Dual Drive feature can realize a large size and high-load system which cannot be realized by a single motor.
- Dual Drive feature suppresses the slave motor status on the current position screen, teaching screen, and so on. By jogging or teaching only of the master motor, both motors move in synchronization.

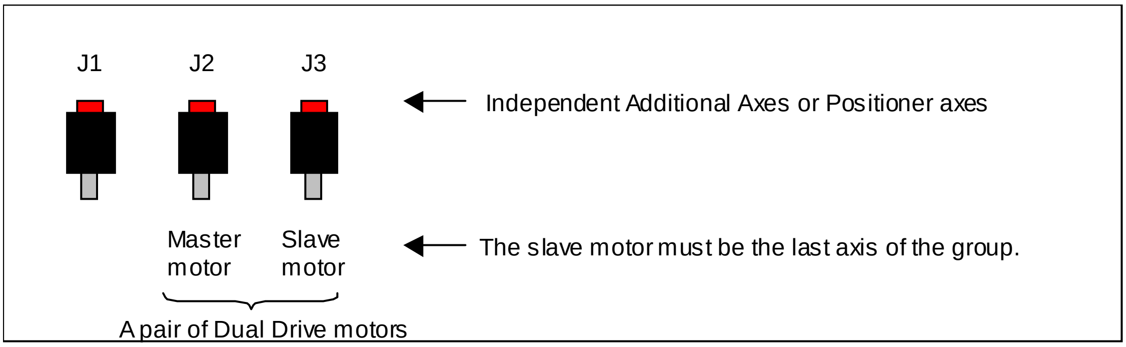
To use this function, Dual Drive function option (A05B-2600-J836) is required. Also only Independent axis (A05B-2600-H895) and Positioner (A05B-2600-H896) can be used as Dual Drive.

18.1 SETUP

Preparation

Setup Independent Additional Axes or Positioner axes which you plan to assign as a pair of Dual Drive motors.

- The pair of Dual Drive motors must belong to the same motion group.
- In this step, setup both the master motor and the slave motor individually.
- The slave motor must be the last axis in the group. If there exist multiple Dual Drive motor pairs in one group, the order of the slave axes is unconfined while they must be the last axes in the group.


J1
J2
J3
Independent Additional Axes or Positioner axes
Master motor
Slave motor
The slave motor must be the last axis of the group.
A pair of Dual Drive motors

Dual Drive motor axes setup

Set the following variables, then cycle power.

System Variable	Description
$SCR_GRP[g].$NUM_AXES	This system variable means the total number of axes in the group. Here, count a pair of Dual Drive motors as one axis. In other words, change this variable to the original number minus the number of Dual Drive motor pairs.
$SCR_GRP[g].$NUM_ROB_AXS	Change this variable in the same manner as $NUM_AXES.
$SCR_GRP[g].$NUM_DUAL	Enter the number of Dual Drive motor pairs. The maximum value of this system variable is 3.
$DUAL_DRIVE[g].$M_AXIS_NUM[n]	Enter the axis number of the master motor. “n” is an index of the Dual Drive motor pair. For the first pair, n = 1.
$DUAL_DRIVE[g].$S_AXIS_NUM[n]	Enter the axis number of the slave motor.

[Example 1]

System configuration:

- Group 2 has three motors as Independent Additional Axes.
- J2-J3 of Group 2 are a pair of Dual Drive motors.
- J2 is the master motor and J3 is the slave motor.

Setting:

- $SCR_GRP[2].$NUM_AXES = (3→) 2
- $SCR_GRP[2].$NUM_ROB_AXS = (3→) 2
- $SCR_GRP[2].$NUM_DUAL = 1
- $DUAL_DRIVE[2].$M_AXIS_NUM[1] = 2
- $DUAL_DRIVE[2].$S_AXIS_NUM[1] = 3

When jogging or teaching, the Dual Drive pair is treated as “J2”.

[Example 2]

System configuration:

- Group 2 has four motors as Independent Additional Axes.
- J1-J3 and J2-J4 of Group 2 are pairs of Dual Drive motors.
- J1 and J2 are the master motors and J3 and J4 are the slave motors.

Setting:

- $SCR_GRP[2].$NUM_AXES = (4→) 2
- $SCR_GRP[2].$NUM_ROB_AXS = (4→) 2
- $SCR_GRP[2].$NUM_DUAL = 2
- $DUAL_DRIVE[2].$M_AXIS_NUM[1] = 1
- $DUAL_DRIVE[2].$M_AXIS_NUM[2] = 2
- $DUAL_DRIVE[2].$S_AXIS_NUM[1] = 3
- $DUAL_DRIVE[2].$S_AXIS_NUM[2] = 4

When jogging or teaching, the Dual Drive pairs are treated as “J1” and “J2”.

19 MULTI UOP INTERFACE FUNCTION

Multi UOP Interface function enable us to use up to 5 UOP sets that have a connection with start of program. Therefore, It is possible to control (select, execute, pause and so on) multiple program independently with a single robot controller.

Specification

The number of usable UOP set
Up to 5 UOP sets are usable.

UOP Interface
The number of UOP signals depend on how many UOP sets is used.

Number of set	1	2	3	4	5
Input	18	23	32	41	50
Output	20	26	36	46	56

Software Options

The following software option is needed for Multi UOP Interface function.

Multi UOP Interface	A05B-2600-J964

If 2 or more motion groups (multiple robot, independent axis and so on) is used, the following software option is needed.

Multi-Group Motion	A05B-2600-J601

Multi UOP Interface for Dual Arm Control (J605)

Multi UOP interface for Dual Arm Control (J605) is the function that is Multi UOP Interface function but it is limited to use UOP sets up to 2. If more than 2 UOP sets are not needed, we can choose not Multi UOP Interface (J964) but Multi UOP interface for Dual Arm Control (J605).

Multi UOP interface for Dual Arm Control is different from Multi Robot Control Function that had been used since R-30*i*A in the following point:

- One CSTOPI signal was prepared for 2 UOP sets by Multi Robot Control Function, but two CSTOPI signals are prepared for 2 UOP sets by Multi UOP interface for Dual Arm Control. (For this difference, Multi UOP interface for Dual Arm Control uses more one signal than Multi Robot Control Function by CSTOPI signal.)
- In case of Multi Robot Control function, the motion groups intended by each UOP depend on coupling structure of Isolation switch. But in case of Multi UOP Interface function, the user can configure the motion groups intended by each UOP freely.

And it is possible to use UOP as the specification of Multi Robot Control function. Please refer to “19.6 System variables”.

19.1 PERIPHERAL I/O

The following signals are provided in response to the number of UOP set.

INPUT SIGNAL	OUTPUT SIGNAL
<Number of set : 1>	
*IMSTP	CMDENBL#1
*HOLD#1	SYSRDY#1
*SFSPD	PROGRUN#1
CSTOPI#1	PAUSED#1
FAULT_RESET	HELD#1
START#1	FAULT#1
HOME	ATPERCH
ENBL	TPENBL
RSR1/PNS1/STYLE1	BATALM
RSR2/PNS2/STYLE2	BUSY
RSR3/PNS3/STYLE3	ACK1/SNO1
RSR4/PNS4/STYLE4	ACK2/SNO2
RSR5/PNS5/STYLE5	ACK1/SNO1
RSR6/PNS6/STYLE6	ACK1/SNO1
RSR7/PNS7/STYLE7	ACK1/SNO1
RSR8/PNS8/STYLE8	ACK1/SNO1
PNSTROBE#1	ACK1/SNO1
PROD_START#1	ACK1/SNO1
	SNACK
	RESERVE
Total 18 points	Total 20 points
<Number of set : 2>	
*HOLD#2	CMDENBL#2
START#2	SYSRDY#2
PNSTROBE#2	HELD#2
PROD_START#2	FAULT#2
CSTOPI#2	PROGRUN#2
	PAUSED#2
Total 23 points	Total 26 points
<Number of set : 3>	
RSR9	ACK9
RSR10	ACK10
RSR11	ACK11
RSR12	ACK12
*HOLD#3	CMDENBL#3
START#3	SYSRDY#3
PNSTROBE#3	HELD#3
PROD_START#3	FAULT#3
CSTOPI#3	PROGRUN#3
	PAUSED#3
Total 32 points	Total 36 points

<Number of set : 4>	
RSR13	ACK13
RSR14	ACK14
RSR15	ACK15
RSR16	ACK16
*HOLD#4	CMDENBL#4
START#4	SYSRDY#4
PNSTROBE#4	HELD#4
PROD_START#4	FAULT#4
CSTOPI#4	PROGRUN#4
	PAUSED#4
Total 41 points	Total 46 points
<Number of set : 5>	
RSR17	ACK17
RSR18	ACK18
RSR19	ACK19
RSR20	ACK20
*HOLD#5	CMDENBL#5
START#5	SYSRDY#5
PNSTROBE#5	HELD#5
PROD_START#5	FAULT#5
CSTOPI#5	PROGRUN#5
	PAUSED#5
Total 50 points	Total 56 points

I/O Configuration:

In case that DI/DO are 40 points

Number of set	1	2	3	4	5
Specialized input（UI）	18	23	32	-	-
General-purpose input（DI）	22	17	8	-	-
Specialized output（UO）	20	26	36	-	-
General-purpose output（DO）	20	14	4	-	-

In case that DI/DO are 96 points

Number of set	1	2	3	4	5
Specialized input（UI）	18	23	32	41	50
General-purpose input（DI）	78	73	64	55	46
Specialized output（UO）	20	26	36	46	56
General-purpose output（DO）	76	70	60	50	40

19.1.1 Setting of the Number of UOP Set

The number of UOP set is 1 by default. Please change $MULTI_ROBO.$NUM_PROG from 1 to the number you want to use when you want to change the number of UOP set. You can change the value up to 5. And the number of UOP set is changed after you cycle power.

Then the number of UI/UO signal is changed, but signal assignment isn't changed. If you clear assignment, new assignment in response to new number of UI/UO signal is given automatically (You can clear assignment by selecting F5, CLR_ASG in I/O Link Device screen. Power cycling is needed after that. And all assignments are cleared. If this is inconvenient for you, please change assignment manually).

Here from, 'N' means the number of UOP set, in other words, the value of $MULTI_ROBO.$NUM_PROG ('N' is used for explanation after "19.1.3 HOLD#1 to #N").

19.1.2 Modification and Addition of Signal

In Multi UOP Interface Function, the meaning of the following signals are changed.

Input:	HOLD#1:	UI[2] HOLD is changed to HOLD#1.
	CSTOPI#1:	UI[4] CSTOPI is changed to CSTOPI#1.
	START#1:	UI[6] START is changed to START#1.
	PNSTROBE#1:	UI[17] PNSTROBE is changed to PNSTROBE#1.
	PROD_START#1:	UI[18] PROD_START is changed to PROD_START#1.
Output:	CMDENBL#1:	UO[1] CMDENBL is changed to CMDENBL#1.
	SYSRDY#1:	UO[2] SYSRDY is changed to SYSRDY#1.
	PROGRUN#1:	UO[3] PROGRUN is changed to PROGRUN#1.
	PAUSED#1:	UO[4] PAUSED is changed to PAUSED#1.
	HELD#1:	UO[5] HELD is changed to HELD#1.
	FAULT#1:	UO[6] FAULT is changed to FAULT#1.

In Multi UOP Interface Function, the following signals are added.

<Number of set : 2>

	Specialized I/O signal	UI/UO[*]
Input	HOLD#2	UI[19]
	START#2	UI[20]
	PNSTROBE#2	UI[21]
	PROD_START#2	UI[22]
	CSTOPI#2	UI[23]
Output	CMDENBL#2	UO[21]
	SYSRDY#2	UO[22]
	HELD#2	UO[23]
	FAULT#2	UO[24]
	PROGRUN#2	UO[25]
	PAUSED#2	UO[26]

<Number of set : 3>

	Specialized I/O signal	UI/UO[*]
Input	RSR9	UI[24]
	RSR10	UI[25]
	RSR11	UI[26]
	RSR12	UI[27]
	HOLD#3	UI[28]
	START#3	UI[29]
	PNSTROBE#3	UI[30]
	PROD_START#3	UI[31]
	CSTOPI#3	UI[32]
Output	ACK9	UO[27]
	ACK10	UO[28]
	ACK11	UO[29]
	ACK12	UO[30]
	CMDENBL#3	UO[31]
	SYSRDY#3	UO[32]
	HELD#3	UO[33]
	FAULT#3	UO[34]
	PROGRUN#3	UO[35]
	PAUSED#3	UO[36]

<Number of set : 4>

	Specialized I/O signal	UI/UO[*]
Input	RSR13	UI[33]
	RSR14	UI[34]
	RSR15	UI[35]
	RSR16	UI[36]
	HOLD#4	UI[37]
	START#4	UI[38]
	PNSTROBE#4	UI[39]
	PROD_START#4	UI[40]
	CSTOPI#4	UI[41]
Output	ACK13	UO[37]
	ACK14	UO[38]
	ACK15	UO[39]
	ACK16	UO[40]
	CMDENBL#4	UO[41]
	SYSRDY#4	UO[42]
	HELD#4	UO[43]
	FAULT#4	UO[44]
	PROGRUN#4	UO[45]
	PAUSED#4	UO[46]

<Number of set : 5>

	Specialized I/O signal	UI/UO[*]
Input	RSR17	UI[42]
	RSR18	UI[43]
	RSR19	UI[44]
	RSR20	UI[45]
	HOLD#5	UI[46]
	START#5	UI[47]
	PNSTROBE#5	UI[48]
	PROD_START#5	UI[49]
	CSTOPI#5	UI[50]
Output	ACK17	UO[47]
	ACK18	UO[48]
	ACK19	UO[49]
	ACK20	UO[50]
	CMDENBL#5	UO[51]
	SYSRDY#5	UO[52]
	HELD#5	UO[53]
	FAULT#5	UO[54]
	PROGRUN#5	UO[55]
	PAUSED#5	UO[56]

19.1.3 HOLD#1 to #N

Here from, the word “Program#1” appears. “Program#1” means the program selected by UOP set 1. “Program#2” to “Program#N” are same as “Program#1”. Please refer to “19.2 SELECT PROGRAM”.

Multi UOP Interface has one HOLD signal per UOP set (#1 to #N). The function of HOLD signal is changed by setting system variable “$MULTI_ROBO.$HOLD_TYPE.

- In case of $MULTI_ROBO.$HOLD_TYPE = 0（Default）
  If whichever of HOLD signals (HOLD#1 to #N) is turned OFF, all the executing programs are paused.

- In case of $MULTI_ROBO.$HOLD_TYPE = 1
  If HOLD#X is turned OFF, “Program#X' is paused (X = 1 to N).
  The running program which is not selected as “Program#X”, can NOT be held. Please turn off ENBL to pause the program.
  (However, if “Program#X” is not selected, when HOLD#X is OFF, all the programs in execution are paused.)

But, when a program is tried to execute, regardless of $MULTI_ROBO.$HOLD_TYPE value, if any HOLD signal is OFF, it is impossible to execute program. Then if program execution is needed, please turn ON all HOLD signals.

19.1.4 CSTOPI#1 to #N

Multi UOP Interface has one CSTOPI signal per UOP set (#1 to #N). The function of CSTOPI signal is changed by setting “CSTOPI for ABORT” or “Abort all programs by CSTOPI” in System Config screen.

- When “Abort all programs by CSTOPI” is FALSE (Default)
  If CSOTPI#X is turned ON, “Program#X” is aborted (X = 1 to N).
  It also release (clear) programs in UOP set X queue from the wait state by RSR.
  Please refer to “19.2.6 RSR”.

- When “Abort all programs by CSTOPI” is TRUE
  If whichever of CSTOPI signals (HOLD#1 to #N) is turned ON, all the executing programs are aborted. It also release (clear) programs in all UOP set queue from the wait state by RSR.

TRUE for “Abort all programs by CSTOPI” is recommended for Multi UOP Interface function.

The way of terminating program is changed by setting “CSTOPI for ABORT” in System Config screen.

- When “CSTOPI for ABORT” is FALSE (Default)
  If CSOTPI signal is turned ON, this signal aborts the program in execution as soon as execution of the program completes.

- When “CSTOPI for ABORT” is TRUE
  If CSOTPI signal is turned ON, this signal immediately aborts the program in execution.

19.1.5 START#1 to #N

Note）

In the system with Multi UOP Interface, multiple programs are controlled at the same time, and it is more complex to control program than control single program. So it is recommended that the Multi UOP Interface is in condition that “START for CONTINUE only:“ in system config screen is “TRUE”. Only paused program can be started by START signal under this setting. In case of starting the ended program, use RSR or PROD_START signal, please.

Multi UOP Interface has one START signal per UOP set (#1 to #N).

“Program#X” is started when START#X is turned off (X = 1 to N).

The program is continued to execute from paused line in case that START signal is input with paused program.

When “START for CONTINUE only:“ in system config screen is “TRUE”, the program, which is aborted (neither running nor paused), can NOT be started and the message “PROG-023 Task is not paused” is displayed in TP screen in case that START signal is inputted with the ended program (neither running nor paused).

19.1.6 RSR

Multi UOP Interface has four RSR signal per UOP set (#1 to #N).
(However, if the number of UOP set is 1, eight RSR is usable.)

Number of set	RSR1 to 4	RSR5 to 8	RSR9 to 12	RSR13 to 16	RSR17 to 20
1	#1		-	-	-
2	#1	#2	-	-	-
3	#1	#2	#3	-	-
4	#1	#2	#3	#4	-
5	#1	#2	#3	#4	#5

And each UOP set has RSR queue, so the number of RSR queue is N.
Please refer to “19.4.1 Robot Service Request (RSR)”.

19.1.7 PNSTROBE#1 to #N,PROD_START#1 to #N

Multi UOP Interface has one PNSTROBE and one PROD_START signal per UOP set (#1 to #N).
Please refer to “19.3.2 Program Number Select (PNS)”.

19.1.8 CMDENBL#1 to #N

Multi UOP Interface has one CMDENBL signal per UOP set (#1 to #N).

CMDENBL#1 is ON if all the following conditions are satisfied.

- Remote condition is satisfied.
- No alarm. (Every FAULT#X are OFF. (X = 1 to N))
- No single step.
- If Isolation switch is used, motion groups that is intended by UOP set 1 are connected.
  (“Remote condition” and “motion groups that is intended by UOP set 1”are mentioned after.)

In case of CMDENBLE#2 to #N, only last condition is different.

- If Isolation switch is used, motion groups that is intended by UOP set 2 (to N) are connected.

1. Remote Condition:
   ・ The Teach Pendant enable switch is set OFF.
   ・ Setting of Remote/Local is set to Remote.
   ・ The *SFSPD input is ON.
   ・ The ENBL input is ON.
   ・ The system variable $RMT_MASTER is 0 (UOP).

2. Motion groups that is intended by UOP set
   Motion groups that is intended by UOP set is changed by setting system variable “$MULTI_ROBO.$MULTI_PROGC”.
   Please refer to “19.2.1 Setting of Motion Group Intended by Each UOP Set”.

19.1.9 SYSRDY#1 to #N

Multi UOP Interface has one SYSRDY signal per UOP set (#1 to #N).

SYSRDY#1 is ON if all the following conditions are satisfied.

- All the axes are ready. (If system has Isolation switch, all the axes except for isolated are ready.)
- If Isolation switch is used, motion groups that is intended by UOP set 1 are connected.

In case of SYSRDY#2 to #N, only last condition is different.

- If Isolation switch is used, motion groups that is intended by UOP set 2 (to N) are connected.

19.1.10 PROGRUN#1 to #N,PAUSED#1 to #N

Multi UOP Interface has one PROGRUN and one PAUSED signal per UOP set (#1 to #N).
PROGRUN and PAUSED signals show the program execution status as the following.

- When program is running:
  PROGRUN#X (X = 1 to N)

・ In case that “Program#X” is in execution.

In case that the program, which is not “Program#X”, is in execution, every PROGRUN#X is output. (Default)

If the value of the system variable $MULTI_ROBO.$UNSELOUTPUT is 0, the behavior of signals is changed as follows.
In case that the program, which is not “Program#X”, is in execution, PROGRUN#X is not output. But BUSY is output.

- When program is paused:
  PAUSED#X (X = 1 to N)
  ・ In case that “Program#X” is paused.

  In case that the program, which is not “Program#X”, is paused, every PAUSED#X is output. (Default)

  If the value of the system variable $MULTI_ROBO.$UNSELOUTPUT is 0, the behavior of signals is changed as follows.
  In case that the program, which is not “Program#X”, is paused, PAUSED#X is not output.

19.1.11 HELD#1 to #N

Multi UOP Interface has one HELD signal per UOP set (#1 to #N).

HELD#X is ON if one of the following conditions is satisfied. (X = 1 to N)

- HOLD#X is OFF.
- The HOLD switch on the Teach Pendant is pushed.
- The HOLD switch on the Operator Panel is pushed.

19.1.12 FAULT#1 to #N

Multi UOP Interface has one FAULT signal per UOP set (#1 to #N).

FAULT#X turns ON when the alarm occurs by “Program#X” execution or other than program execution. (X = 1 to N)

When system is in alarm status, at least one of FAULT#X must be ON.

19.1.13 ATPERCH

The specified signal ATPERCH is set to ON when the robot of motion group 1 is at the reference position 1. Please assign reference position output except motion group 1.

19.2 SELECT PROGRAM

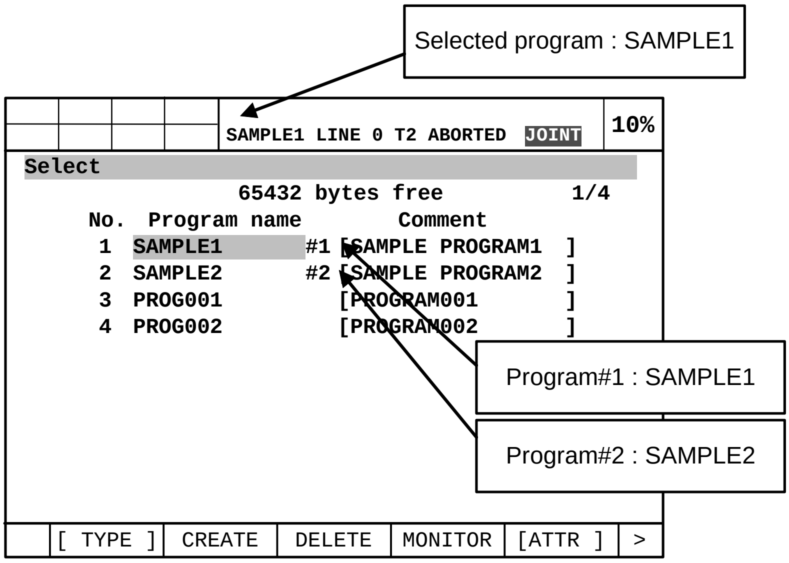
In Multi UOP Interface function, it is possible for each UOP set to select program individually.
One UOP set is given one program # for selected program.

Here from, “Program#1” means the program selected by UOP set 1.

“Program#1” is run by UOP START#1, PROD_STAR#1.
On program selection screen, “#1” is displayed to the right of program name that is used as “Program #1”.

“Program#2” to “Program#N” are same as “Program#1”.

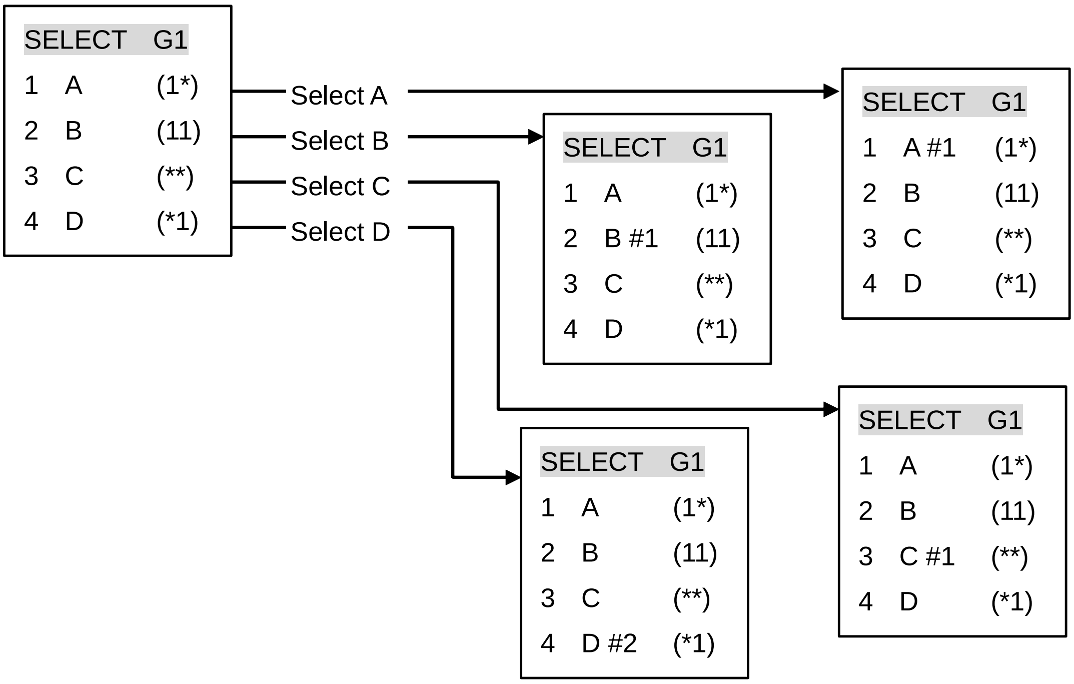
On the contrary, a program that is displayed on program edit screen is referred as “selected program”.
[FWD]/[BWD] keys on teach pendant or start button on Operator Panel run “selected program”.
“Selected program” is displayed on left side of second line of TP screen.



```
                    SAMPLE1 LINE 0 T2 ABORTED  JOINT   10%
Select
                   65432 bytes free          1/4
     No.  Program name        Comment
      1  SAMPLE1        #1 [SAMPLE PROGRAM1  ]
      2  SAMPLE2        #2 [SAMPLE PROGRAM2  ]
      3  PROG001           [PROGRAM001       ]
      4  PROG002           [PROGRAM002       ]

    [ TYPE ]  CREATE   DELETE   MONITOR  [ATTR ]  >
```

If you set cursor to a program and presses ENTER, “selected program” is changed to the program.
Then, the program changes to “Program#X” automatically (What # is set depend on motion group of the program and what motion group is intended by each UOP sets).

For example, when the number of UOP sets is 2, and UOP set 1 intends Motion group 1 and UOP set 2 intends Motion group 2, the behavior is as the following figure.



(1*)	The program has only group 1.
(*1)	The program has only group 2.
(11)	The program has group 1 and 2.

19.2.1 Setting of Motion Groups Intended by Each UOP Set

What UOP set intends what motion group, in other words, what # is set to a program is changed by setting system variable "$MULTI_ROBO.$MULTI_PROGC.

$MULTI_PROGC has 5 alignments, and each one corresponds one UOP set. If the number of UOP set is smaller than 5, $MULTI_PROGC alignments are available as many as the number of UOP set, and the other alignments is ineffective.

UOP set 1: $MULTI_PROGC[1]

UOP set 2: $MULTI_PROGC[2]

UOP set 3: $MULTI_PROGC[3]

UOP set 4: $MULTI_PROGC[4]

UOP set 5: $MULTI_PROGC[5]

The each bit of $MULTI_PROGC value is equivalent to each motion group.

Bit31	・・・	Bit1	Bit0
Motion Group32	・・・	Motion Group2	Motion Group1

For example, the default values of $MULTI_PROGC are set as the following, it is assumed that one UOP set is equivalent to one motion group.

Example1 (Default)

UOP set 1: $MULTI_PROGC[1] = 1

UOP set 2: $MULTI_PROGC[2] = 2

UOP set 3: $MULTI_PROGC[3] = 4

UOP set 4: $MULTI_PROGC[4] = 8

UOP set 5: $MULTI_PROGC[5] = 16

In case that PROGRAM1 is equivalent to motion group 1,4,
PROGRAM2 is equivalent to motion group 2,5,
and PROGRAM3 is equivalent to motion group 3,
$MULTI_PROGC value is the following.

Example 2

UOP set 1: $MULTI_PROGC[1] = 9

UOP set 2: $MULTI_PROGC[2] = 18

UOP set 3: $MULTI_PROGC[3] = 4

UOP set 4: $MULTI_PROGC[4] = 0

UOP set 5: $MULTI_PROGC[5] = 0

Every UOP set can select a program having no motion group. However, when a program having no motion group is selected by TP, if $MULTI_PROGC[#x] is 0, the program become "Program#X" by priority.

Detailed explanation is mentioned in "19.6 SYSTEM VARIABLES". Please refer it.

19.2.2 Select Program at Running or Paused

When “Program#1” is paused, if you try to select another program as “Program#1”:

- When program is selected by UOP, you can not select the program, and the message “TPIF-013 Other program is running” is displayed.
- When program is selected by Teach Pendant and Teach Pendant Enable switch is OFF, you can not select the program, and the message “TPIF-013 Other program is running” is displayed.
- When program is selected by Teach Pendant and Teach Pendant Enable switch is ON, the original “Program#1” is aborted, and the selected program becomes “Program#1”.(This is the case that “Multi Program Selection” is disabled. If “Multi Program Selection” is enabled, “Program#1” doesn’t change.)

When “Program#1” is running, if you select another program as “Program#1”, you can not select the program regardless of selection way, and the message “TPIF-013 Other program is running” is displayed.

“Program#2” to “Program#N” are same as “Program#1”.

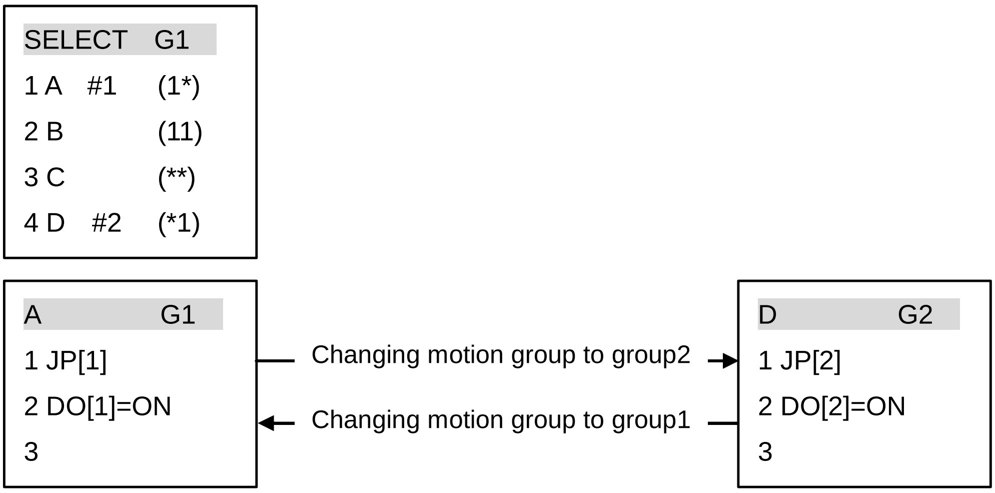
19.2.3 Program Edit Screen Display When Operating Motion Group is Changed

When the user teach motions to each motion group, the user needs to operate to select motion group that can be operated by teach pendant.
It is possible to select motion group to select “CHANGE GROUP” in function menu.

Currently selected motion group is displayed on teach pendant like “G1”,”G2”.

By changing motion group, “selected program” is also changed according to selected motion group. Changed ”selected program” is displayed on program edit screen.

For example, UOP set 1 intends Motion group 1, and UOP set 2 intends Motion group 2.
Suppose “Program #1 “ has only group1 and “Program #2 has” only group2.
By Selecting group1, “Program #1”
By Selecting group2, “Program #2”
automatically becomes “selected program” and the program is displayed on program edit screen.



```
SELECT    G1
1 A   #1    (1*)
2 B         (11)
3 C         (**)
4 D   #2    (*1)
```

```
A          G1
1 JP[1]
2 DO[1]=ON
3
```

```
D          G2
1 JP[2]
2 DO[2]=ON
3
```

Changing motion group to group2
Changing motion group to group1

If “Selected program” has both group1 and group2 or “selected program” has no motion group, change of motion group doesn’t change “selected program”.

To change “selected program” in these cases, display program selection screen, set cursor to a program that you want to display and press ENTER.

19.3 RUN PROGRAM

19.3.1 Robot Service Request (RSR)

If the number of UOP set is 1, when RSR1 to 8 are turned on, the program corresponded to the RSR number is started if the program is aborted.
If another program is running or paused, the request is joined to the queue. And the program is started when the running program is completed.

Multi UOP Interface has four RSR signals per UOP set (#1 to #N) if the number of UOP set is 2 and over. RSR1 to 4 are for UOP set 1. RSR5 to 8, RSR9 to 12, … are same as RSR1 to 4 (Please refer to “19.2.6 RSR”.). And each UOP set has RSR queue, so the number of RSR queue is N.
By this function, N programs can run at the same time.

The program started by RSR1 to 4 becomes “Program#1”.

In the following situation, program is not started.

- The program started by RSR1 to 4 does not have motion group intended by UOP set 1 and it has motion group intended by other UOP set.

In this case, message “TPIF-44 Program is unsuitable for robot” is displayed.

RSR5 to 8, RSR9 to 12, … are same as RSR1 to 4.

19.3.2 Program Number Select (PNS)

Multi UOP Interface has one PNSTROBE and one PROD_START signal per UOP set (#1 to #N).

The program selected by PNSTROBE#X becomes “Program#X”. This program can be started by PROD_START#X and it is resumed from pause by START#X (X = 1 to N).

In the following situation, program is not selected.

- The program selected by PNSTROBE#X does not have motion group intended by UOP set X and it has motion group intended by other UOP set.

In this case, message “TPIF-44 Program is unsuitable for robot” is displayed.

19.4 STOP PROGRAM

19.4.1 Pause Program by UOP

When whichever of HOLD signals (HOLD#1 to #N) is turned OFF, running program is paused.
Please read “19.1.3 HOLD#1 to #N” for detail.

19.4.2 Abort Program by UOP

If “CSTOPI for ABORT” in System Config is TRUE, programs are aborted when CSTOPI turns ON.

The default setting of 'CSTOPI for ABORT' is FALSE.
Please read "19.1.4 CSTOPI#1 to #N" for detail.

19.4.3 Other

When hold button on Teach Pendant or Operator Panel, all running programs are paused.

19.5 SYSYTEM VARIABLES

Name Type Protection Range	$MULTI_ROBO.$MLT_ENABLE BOOLEAN Protected TRUE/FALSE
Meaning	Show whether Multi UOP Interface function is enabled or not. TRUE: Multi UOP Interface function is enabled. FALSE: Multi UOP Interface function is disabled.

Name Default Type Protection Range	$MULTI_ROBO.$HOLD_TYPE 0 INTEGER Not protected 0/1
Meaning	Select the function of HOLD#1 to #N. 0: All programs are paused when one of the HOLD#1 to #N signals turns OFF. 1: "Program#X" is paused when HOLD#X turns OFF.

Name Default Type Protection Range	$MULTI_ROBO.$NUM_PROG 1 INTEGER Not protected 1 to 5
Meaning	The number of UOP set to use. Set this value to 1 if you want to disable Multi UOP Interface function.

Name Default Type Protection Range	$MULTI_ROBO.$MULTI_PROGC[5] [1]1, [2]2, [3]4, [4]8, [5]16 INTEGER Not protected 0 to 0xFFFF
Meaning	Bit mask for intended motion group of each UOP set. In the default, UOP set 1 intends Motion group 1, UOP set 2 intends Motion group 2, and so on.

[Supplement] Basis of selection for program #

We can select program by UOP or TP, and program number is set on the basis of $MULTI_PROGC. Program number setting rule is the following:

< Common rule for UOP and TP >

Rule 1) If motion group of $MULTI_PROGC[#n] and motion group of program match at least one group, the program can be set #n.

Rule 2) No motion group program can be set ALL program number.

Rule 3) If motion group of every $MULTI_PROGC[#n] and motion group of program match none, the program is treated equally as no motion group program. (So conform to Rule 2.)
Example $MULTI_PROGC[1-5] = 1, 2, 3, 4, 0

	Motion group of program A to G	Program number				
	G1 , 2 , 3 , 4 , 5 , 6 , 7 , 8	#1	#2	#3	#4	#5
PROGRAM A	(1 , * , * , * , * , * , * , *)	#1		#3		
PROGRAM B	(* , 1 , * , * , * , * , * , *)		#2	#3		
PROGRAM C	(1 , 1 , * , * , * , * , * , *)	#1	#2	#3		
PROGRAM D	(1 , * , 1 , * , * , * , * , *)	#1		#3	#4	
PROGRAM E	(* , * , * , * , * , * , * , *)	#1	#2	#3	#4	#5
PROGRAM F	(* , 1 , * , 1 , * , * , * , *)		#2	#3		
PROGRAM G	(* , * , * , 1 , * , * , * , *)	#1	#2	#3	#4	#5

The above figure illustrates PROGRAM C can be set #1, #2, or #3, and conversely, #2 can select PROGRAM B, C, E, F or G, for instance.

$MULTI_PROGC[1] = 1 >> match at least one group to PROGRAM A, C, D
$MULTI_PROGC[2] = 2 >> match at least one group to PROGRAM B, C, F
$MULTI_PROGC[3] = 3 >> match at least one group to PROGRAM A, B, C, D, F
$MULTI_PROGC[4] = 4 >> match at least one group to PROGRAM D
$MULTI_PROGC[5] = 0 >> match at least one group to none

$MULTI_PROGC[1-5] >> can select no motion group PROGRAM E, G (along Rule 3)

< Rule for TP only >
Rule 4) If selected program is already set program number, program number don't change.
Rule 5) If selected program is not set program number yet, the program is set program number along the following priority. And if some program number have same priority, small number comes first. (In the following Priorities from 1 to 4, #X is the number that "Program #X" is aborted or is not selected.)

[Priority 1] Motion group of $MULTI_PROGC[#x] is equal to motion group of program. (If PROGRAM C is selected by TP, set #3 not #1 or #2. If PROGRAM E is selected by TP, set #5 not #1, #2, #3 or #4.)
[Priority 2] Motion group of $MULTI_PROGC[#x] inhere motion group of program. (If PROGRAM B is selected by TP, set #2 not #3.)
[Priority 3] Motion group of $MULTI_PROGC[#x] and motion group of program match at least one group.
[Priority 4] If program isn't along the above priority from 1 to 3, the program is set to smallest number X that "Program #X" is aborted or is not selected. (A program along Rule 2 or 3 is treated here unless along priority 1.)
[Priority 5] #X is the number that "Program#X" is paused. On that basis, the smallest number that complies with priority 3. (Priority 5 is applied when TP is enabled and "Multi Program Selection" is disabled. Then paused program is aborted.)

When none of the above Priorities is corresponded, it means that all "Program#X" that #X complies with Priorities from 1 to 4 are in execution. Then warning message is displayed and # isn't set to selected program.

Name	$MULTI_ROBO.$TRAD_MODE
Default	FALSE
Type	BOOLEAN
Protection	Not protected
Range	TRUE/FALSE
Meaning	Compatible flag for Multi Robot Control Function used in the old robot controller. TRUE: Compatible mode - There is single CSTOPI signal. - $MULTI_ROBO.$NUM_PROG is 2. - $ROBOT_ISOLC plays role of $MULTI_ROBO.$MULTI_PROGC. FALSE: Normal Multi UOP Interface mode

Name	$MULTI_ROBO.$UNSELOUTPUT
Default	1
Type	INTEGER
Protection	Not protected
Range	0/1
Meaning	Output form of PROGRUN#1 to #N signals at the execution of an unselected program and PAUSE#1 to #N signals at the pause of an unselected program. 0: When an unselected program is executed, PROGRUN#1 to #N signals are not output, but BUSY is output. When an unselected program is paused, PAUSE #1 to #N signals are not output. 1: When an unselected program is executed, all PROGRUN#1 to #N signals are output. When an unselected program is paused, all PAUSE #1 to #N signals are output.

20 ERROR CODE OUTPUT FUNCTION

To use this function, Error Code Output Function Option (J527) and expanded I/O board are required. This function outputs the contents of the alarm that occurred in the robot as a digital output signal code with strobe.

20.1 SPECIFICATION

20.1.1 Types of Alarms

All alarms except ones whose severity WARN are the targets of external output.

20.1.2 Input and Output Signals

Expanded I/O board is necessary to use this function.
33 points of DO are used as output signals of external error code, and 1 point of DI is used as input signal.

Set the signals used in this function to the following system variables.

$ER_OUT_PUT.$OUT_NUM : **Output signal number**
$ER_OUT_PUT.$IN_NUM : **Input signal number**

- Output signals
  $ER_OUT_PUT.$OUT_NUM determines which DO is used.
  33 points of DO from setup number are used.
  When setup number is 0, this function is disabled.

- Types of output signal
  Output of both code signals and strobe are kept on until reset signal is inputted.

Table 20.1.2 Meanings of DO (in case of $ER_OUT_PUT.$OUT_NUM=1)

DO	Types of alarms
1 to 32	Output alarms as binary (32 bit).
33	Used as strobe.

- Input signal (1bit)
  When there are multiple errors, input signal is used as search signal.
  $ER_OUT_PUT.$IN_NUM determines which DI is used.

- Time chart of signals
  When an alarm occurs, the time chart is as follows.

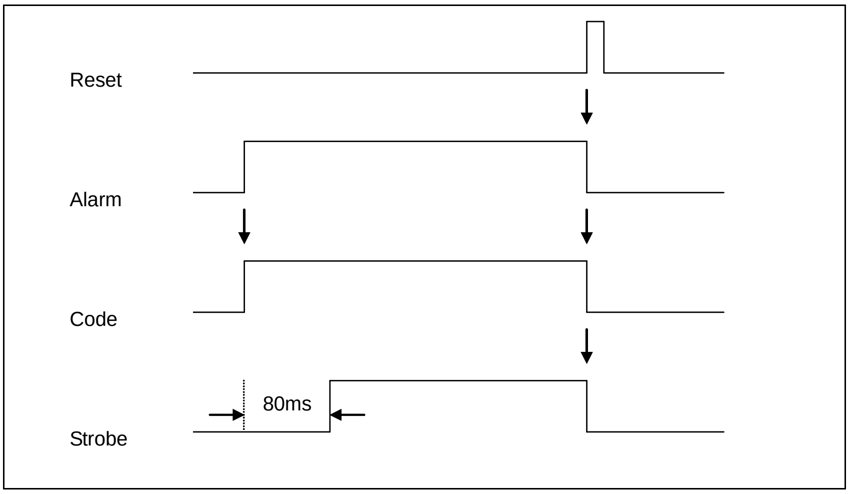


Fig. 20.1.2 (a) Time chart of an alarm

When multiple alarms occur at a time, the initial alarm is outputted at first. Then, the next alarm is output every time search signal is input. These are cyclically output and after that reset signal with strobe is output. Up to 100 alarms can be stored.

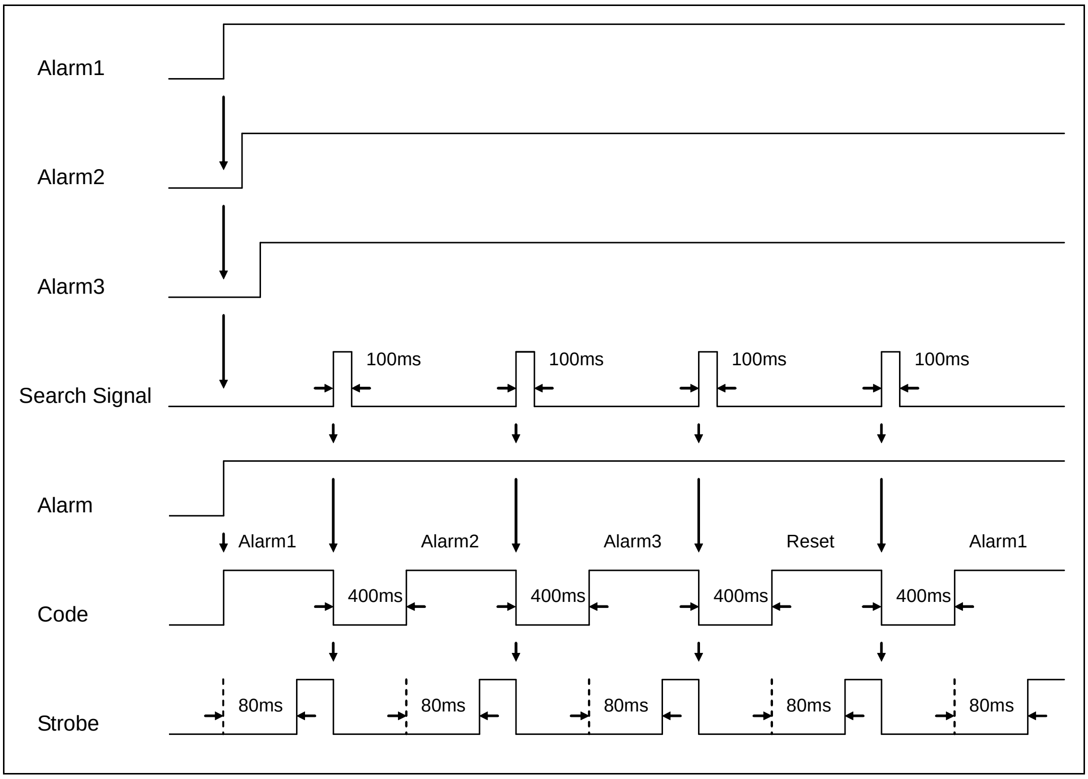


Fig. 20.1.2 (b) Time chart of multiple alarms

20.2 MEANING OF ALARM CODE

In case of $ER_OUT_PUT.$OUT_NUM = 1

DO	Types of alarms
1 to 16	Output an alarm number as binary (16bit).
17 to 24	Output an alarm ID as binary (8bit).
25 to 32	Output an alarm Severity as binary (8bit).
33	Used as strobe.

20.2.1 Severity of Alarm

DO Severity	25	26	27	28	29	30	31	32
NONE	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
WARN	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

Errors that have the severity above will not be outputted.

DO Severity	25	26	27	28	29	30	31	32
PAUSE.L	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF
PAUSE.G	OFF	ON	OFF	OFF	OFF	ON	OFF	OFF
STOP.L	OFF	ON	ON	OFF	OFF	OFF	OFF	OFF
STOP.G	OFF	ON	ON	OFF	OFF	ON	OFF	OFF
SERVO	OFF	ON	ON	OFF	ON	ON	OFF	OFF
ABORT.L	ON	ON	OFF	ON	OFF	OFF	OFF	OFF
ABORT.G	ON	ON	OFF	ON	OFF	ON	OFF	OFF
SERVO2	ON	ON	OFF	ON	ON	ON	OFF	OFF
SYSTEM	ON	ON	OFF	ON	ON	ON	ON	OFF

DO[25] and DO[26] means the severity of running a program.

DO［25］	DO［26］	
OFF	OFF	The program continues running.
OFF	ON	The program will be paused.
ON	ON	The program will be aborted.

DO[27] and DO[28] means the severity of motion.

DO［27］	DO［28］	
OFF	OFF	The motion continues running.
ON	OFF	The motion will be stopped.
OFF	ON	The motion will be canceled.

DO[29] means the severity of a servomotor.

OFF	The servomotor continues running.
ON	The servomotor will be powered down.

DO[30] means the severity of the target tasks.

OFF	Effective only for the task.
ON	Effective for all the tasks.

DO[31] means the request for recovery.

OFF	No need to power down.
ON	Cold start is necessary.

DO[32] means the request for the display on TP.

OFF	No need to display.
ON	Need to display.

20.2.2 Alarm ID

Alarm ID will be outputted as the number in the following.

Typical alarm ID:

Number	Alarm ID
0	OS
3	PROG
7	MEMO
9	TPIF
11	SRVO
12	INTP
15	MOTN
19	JOG
20	APPL
23	SPOT
24	SYST
26	PALT
53	ARC
57	MACR
58	SENS
59	COMP

Other alarm ID:

Number	Alarm ID
1	SRIO
2	FILE
4	COND
5	ELOG
6	MCTL
8	GUID
10	FLPY
13	PRIO
14	TPAX
16	VARS
17	ROUT
18	WNDW
21	LANG
25	SCIO
27	UAPL
33	DICT
34	KCLI
35	TRAN

Number	Alarm ID
36	TKSP
37	COPT
38	APSH
42	CMND
43	RPM
44	LNTK
45	WEAV
46	TCPP
47	TAST
48	MUPS
49	MIGE
50	LSR
51	SEAL
52	PANE
54	TRAK
55	CALB
56	SP
60	THSR

20.2.3 Alarm Number

Alarm number will be directly output as binary data.

Example

In case of “SRVO 002 (severity SERVO)”
Alarm number is 2.
Alarm ID “SRVO” is 11.
The severity is SERVO.

DO ［1］	OFF	Alarm number 2
DO ［2］	ON	
DO ［3］	OFF	
DO ［4］	OFF	
DO ［5］	OFF	
DO ［6］	OFF	
DO ［7］	OFF	
DO ［8］	OFF	
DO ［9］	OFF	
DO ［10］	OFF	
DO ［11］	OFF	
DO ［12］	OFF	
DO ［13］	OFF	
DO ［14］	OFF	
DO ［15］	OFF	
DO ［16］	OFF	

DO ［17］	ON	Alarm ID 11
DO ［18］	ON	
DO ［19］	OFF	
DO ［20］	ON	
DO ［21］	OFF	
DO ［22］	OFF	
DO ［23］	OFF	
DO ［24］	OFF	
DO ［25］	OFF	Alarm Severity SERVO
DO ［26］	OFF	
DO ［27］	ON	
DO ［28］	OFF	
DO ［29］	ON	
DO ［30］	ON	
DO ［31］	OFF	
DO ［32］	OFF	

21 DATA MONITOR FUNCTION

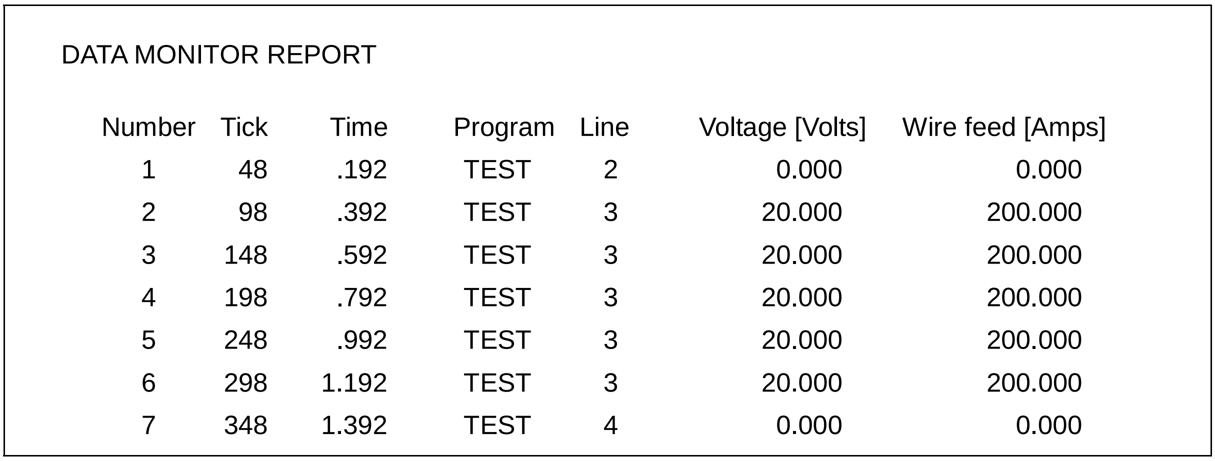
Data Monitor is a tool for improving process quality. You can use it to monitor and record important process parameters. It can alert you to a parameter going out of limit and it can record data for use in a quality record. To use this function, data monitor option (J675) is required.

Data Monitor operates much like a strip chart recorder or a data acquisition system. To use the Data Monitor feature you make selections in two teach pendant screens and add two teach pendant program instructions. Specifically,

- Enable or disable any specific features in the Data Monitor Utility screen.
- Select items to monitor (such as arc current feedback) with one of the Data Monitor schedules.
- Add Sample Start[schedule number] and Sample End instructions to your TP program to control when monitoring occurs.

You can monitor up to six items at once with a Data Monitor schedule. The maximum sampling frequency is 250 Hz. You can specify separate frequencies for limit checking and for recording. As the items specified in the schedule are recorded, the following data is also collected: time, date, distance, program name and line number. You can choose the items you want to monitor from the Data Monitor Schedule screen.

As data is recorded, it can be formatted as a report and sent to a file.
A short example report is shown in Fig.21 (a). The data is tab delimited for importing into a spreadsheet application.



```
DATA MONITOR REPORT

   Number  Tick    Time    Program  Line    Voltage [Volts]   Wire feed [Amps]
      1      48    .192     TEST      2            0.000              0.000
      2      98    .392     TEST      3           20.000            200.000
      3     148    .592     TEST      3           20.000            200.000
      4     198    .792     TEST      3           20.000            200.000
      5     248    .992     TEST      3           20.000            200.000
      6     298   1.192     TEST      3           20.000            200.000
      7     348   1.392     TEST      4            0.000              0.000
```

Fig.21 (a) Report example

Definitions

This section contains definitions of terms you should know to use Data Monitor.

Item - A specific data element to be monitored. For example, an I/O signal, like WO[2] or AI[2]. Data Monitor can monitor the following kinds of items:

- System variables (Real or Integer only)
- KAREL program variables (Real or Integer only)
- I/O ports (digital and analog)
- Registers (numeric only)
- Axis position

Schedule -A set of parameters that define how to monitor specific items and where to save recorded data.
Trigger - A condition that must be met to begin or end monitoring.
Limit - A defined high or low value for a monitored item.

Monitoring Limits

Data monitoring can check each sampled item against upper and lower limits. If the average of the item samples is out of limits for a specified time period (Tmin in Figure 21 (b)), a warning or a pause alarm will occur, if enabled. You must specify a nominal value, a warning limit, a pause limit, and a time duration for each monitored data item. A warning limit and a pause limit are indicated by the difference from nominal value. If the monitored data exceeds the range of (nominal value + limit value), an alarm is posted.

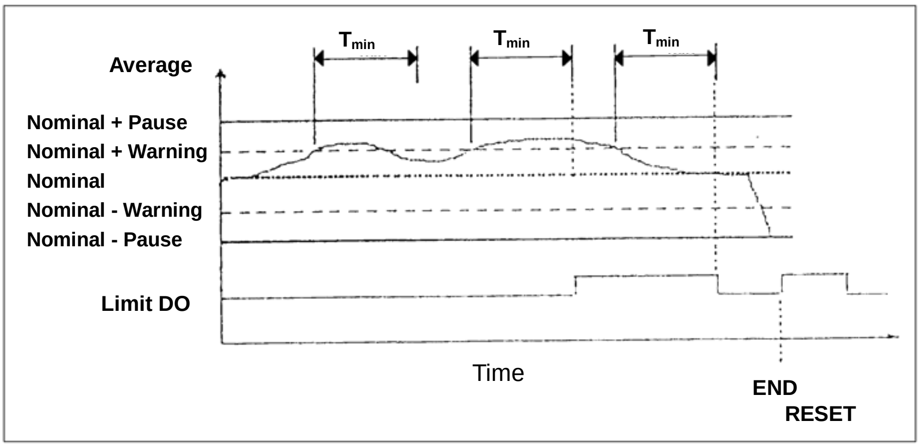


Fig.21 (b) Process limits

If a warning limit is crossed for the specified time, a WARN severity error is posted and the limit digital output is turned ON. If the average of the item samples returns to within the WARN limit for the specified time the digital output is turned OFF.

If a pause limit is crossed for the specified time a PAUSE severity error is posted, the limit digital output is set to ON, and the program is paused.

When the program ends, if a WARN or PAUSE limit error occurred during execution, the limit digital output is turned ON. It is turned OFF by a system RESET.

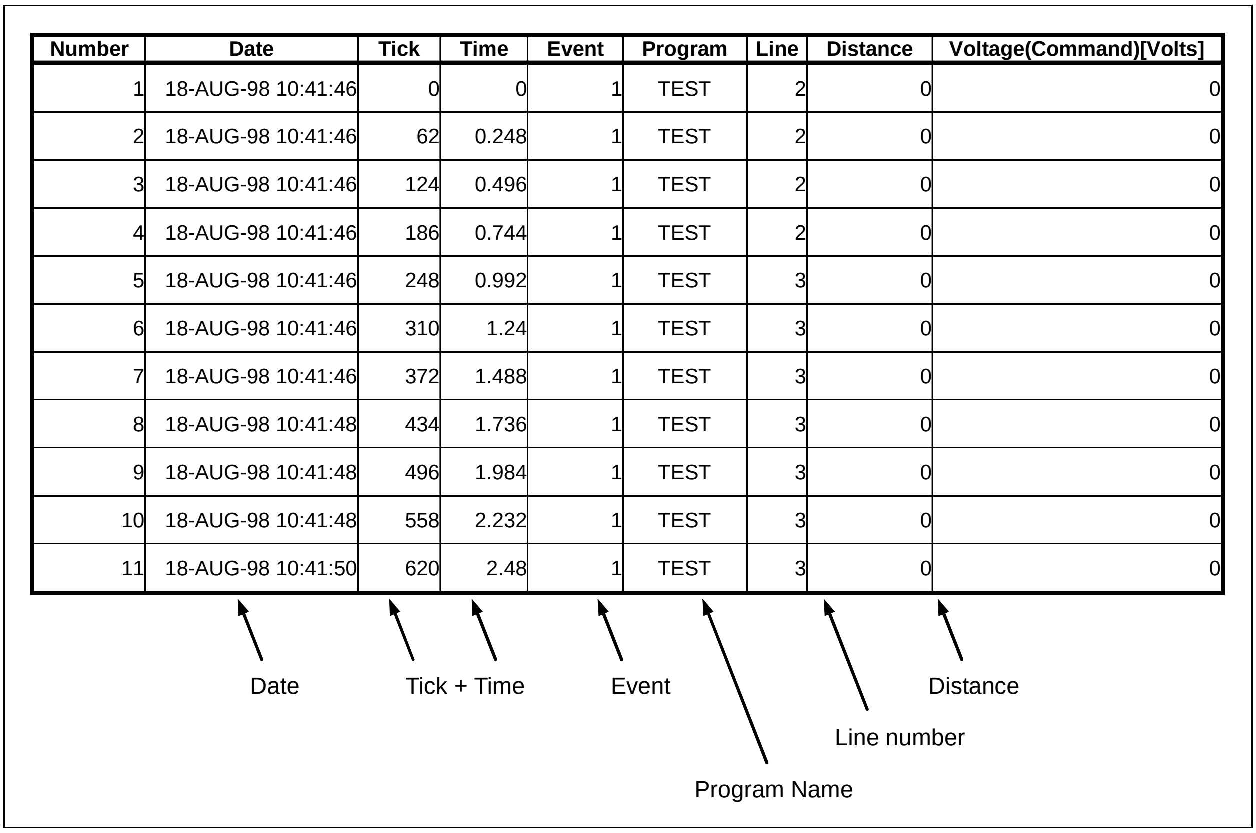
21.1 DATA MONITOR SETUP

You must set up Data Monitor before you can use it. Table 21.1 lists and describes the items found on the Data Monitor Setup screen.

Table 21.1 Data monitor setup screen menu items

ITEM	DESCRIPTION
Data Monitor Operation Default: Enabled	This item enables and disables operation of the Data Monitor function. NOTE Data monitoring is also disabled during step mode.
Recording Default: Enabled	This item turns data recording ON and OFF.
Filing Default: Enabled	This item turns report filing ON and OFF.
Pause on File Errors Default: Disabled	This item controls the severity of certain file errors. If this item is disabled, WARNING severity errors are posted and execution of the program being monitored will continue. If it is enabled, PAUSE severity errors are posted and execution of the program being monitored stops.
Warning Limits Default: Disabled	This item turns Warning Limits ON and OFF.
Pause Limits Default: Disabled	This item turns Pause Limits ON and OFF.

ITEM	DESCRIPTION
Limit Error Output Default: DO[0]	This item defines the port type and port number for the limit output. This digital output is turned ON when a limit error is detected.
Sample Buffer Size Default:10 Min:1 Max:99	This item specifies the size of the sample buffer. Usually you do not have to change the default value.
Record Buffer Size Default:10 Min:1 Max:99	This item specifies the size of the record buffer. If “ONE BUFFER” is specified as record mode (refer table 21.2 Data monitor menu items), this item should be number of data you want to record. (Recording will be finished when record buffer become full.) if “CONTINUOUS” is specified as record mode, you don’t have to change the default value.
Setup Default: Disabled	This item enables or disables printing of data monitor setup information in the report header.
Items Default: Disabled	This item enables or disables printing of information in the report header about each of the items you want to monitor.
Schedule Default: Disabled	This item enables or disables printing of Schedule information in the report header.
Triggers Default: Disabled	This item enables or disables printing of Trigger information in the report header.
Program Name Default: Enabled	This item enables or disables printing the program name column in the Data Monitor report. See Fig. 21.1.
Line Number Default: Enabled	This item enables or disables printing the line number column in the Data Monitor report. See Fig. 21.1.
Date Default: Disabled	This item enables or disables printing the date and time of day column in the Data Monitor report. See Fig. 21.1.
Tick + time Default: Disabled	This item enables or disables printing the Tick and Time column in the Data Monitor report. See Fig. 21.1.
Event Default: Disabled	This item enables or disables printing the event column in the Data Monitor report. See Fig. 21.1. NOTE At this time, there is only one event defined. An event value of 1 indicates the data was recorded as a result of the recording frequency.
Distance Default: Enabled	This item enables or disables printing the distance column in the Data Monitor report. See Fig. 21.1.



Number	Date	Tick	Time	Event	Program	Line	Distance	Voltage(Command)[Volts]
1	18-AUG-98 10:41:46	0	0	1	TEST	2	0	0
2	18-AUG-98 10:41:46	62	0.248	1	TEST	2	0	0
3	18-AUG-98 10:41:46	124	0.496	1	TEST	2	0	0
4	18-AUG-98 10:41:46	186	0.744	1	TEST	2	0	0
5	18-AUG-98 10:41:46	248	0.992	1	TEST	3	0	0
6	18-AUG-98 10:41:46	310	1.24	1	TEST	3	0	0
7	18-AUG-98 10:41:46	372	1.488	1	TEST	3	0	0
8	18-AUG-98 10:41:48	434	1.736	1	TEST	3	0	0
9	18-AUG-98 10:41:48	496	1.984	1	TEST	3	0	0
10	18-AUG-98 10:41:48	558	2.232	1	TEST	3	0	0
11	18-AUG-98 10:41:50	620	2.48	1	TEST	3	0	0

Fig. 21.1 Sample report

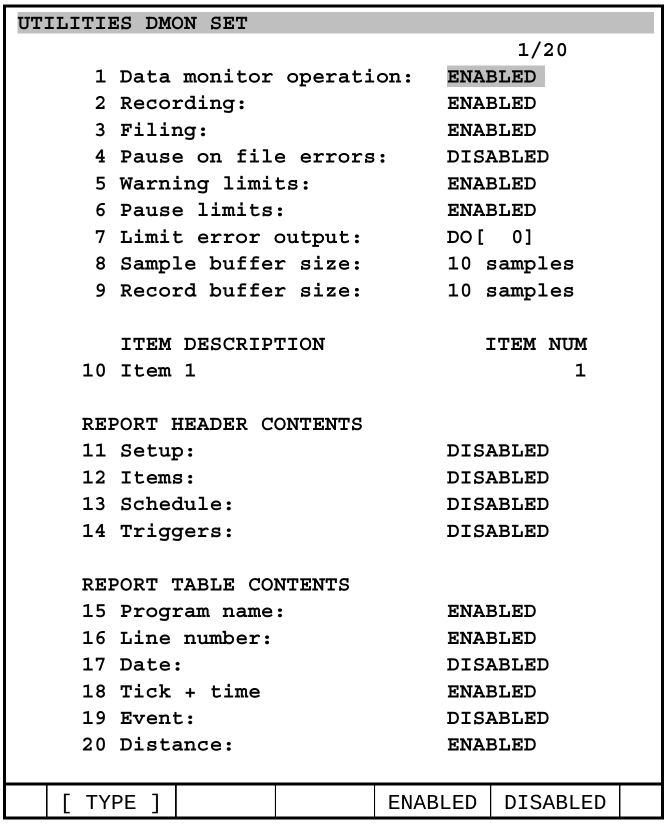
Procedure 21-1 Setting up data monitor

Condition

Data Monitor is installed on your controller.

Step

1 Press [MENU] key.
2 Select UTILITIES.
3 Press F1, [TYPE].
4 Select Data Monitor. The following screen will be displayed.



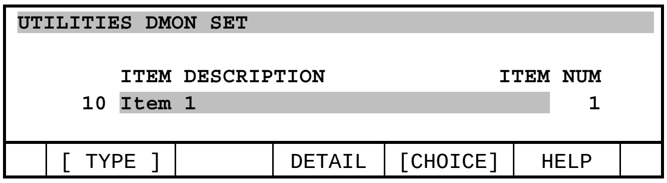
```
UTILITIES DMON SET
                                          1/20
      1 Data monitor operation:   ENABLED
      2 Recording:                ENABLED
      3 Filing:                   ENABLED
      4 Pause on file errors:     DISABLED
      5 Warning limits:           ENABLED
      6 Pause limits:             ENABLED
      7 Limit error output:       DO[  0]
      8 Sample buffer size:       10 samples
      9 Record buffer size:       10 samples

        ITEM DESCRIPTION             ITEM NUM
     10 Item 1                              1

     REPORT HEADER CONTENTS
     11 Setup:                    DISABLED
     12 Items:                    DISABLED
     13 Schedule:                 DISABLED
     14 Triggers:                 DISABLED

     REPORT TABLE CONTENTS
     15 Program name:             ENABLED
     16 Line number:              ENABLED
     17 Date:                     DISABLED
     18 Tick + time               ENABLED
     19 Event:                    DISABLED
     20 Distance:                 ENABLED

  [ TYPE ]                   ENABLED  DISABLED
```

5 Select each item on the menu and set it as desired.

Edit Data Monitor Items

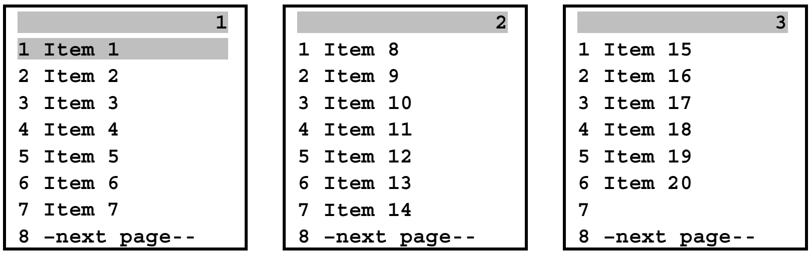
6 Data Monitor allows you to define 20 items to monitor. These items are initialized for you, but you can edit them to suit your needs. If you want to edit Data Monitor items, move the cursor to item 10. When the cursor is on item 10, the DETAIL, [CHOICE], and HELP function keys will be available. [CHOICE] allows you to choose an item from a list. DETAIL allows you to edit that item.

Item 10 has two columns. The right column contains an item number from 1 to 20. The left column contains the corresponding item description.



```
UTILITIES DMON SET

        ITEM DESCRIPTION               ITEM NUM
     10 Item 1                                1

  [ TYPE ]          DETAIL  [CHOICE]   HELP
```

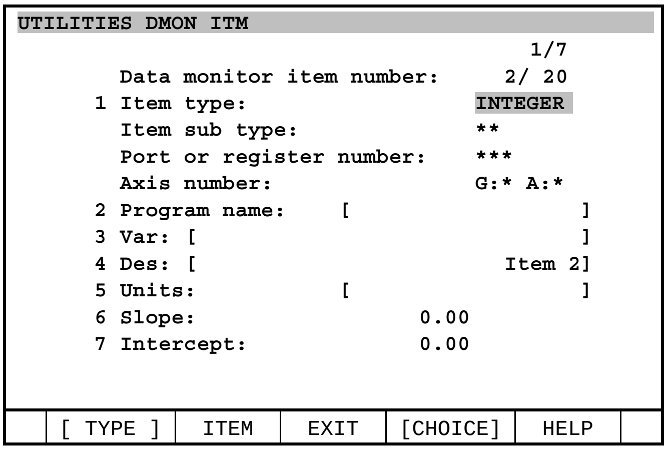
7 Press F4, [CHOICE]. The following screen will be displayed.



```
                1                     2                     3
1 Item 1              1 Item 8              1 Item 15
2 Item 2              2 Item 9              2 Item 16
3 Item 3              3 Item 10             3 Item 17
4 Item 4              4 Item 11             4 Item 18
5 Item 5              5 Item 12             5 Item 19
6 Item 6              6 Item 13             6 Item 20
7 Item 7              7 Item 14             7
8 -next page--        8 -next page--        8 -next page--
```

8 Select an item from the list by number or by moving the cursor and pressing ENTER.

9 Press the F3, DETAIL function key to edit the selected item. In this example, [2 Item 2] has been selected. You will see a screen similar to the following.



```
UTILITIES DMON ITM
                                              1/7
        Data monitor item number:           2/ 20
      1 Item type:                        INTEGER
        Item sub type:                    **
        Port or register number:          ***
        Axis number:                      G:* A:*
      2 Program name:      [                   ]
      3 Var: [                                 ]
      4 Des: [                             Item 2]
      5 Units:             [                   ]
      6 Slope:                      0.00
      7 Intercept:                  0.00

   [ TYPE ]   ITEM    EXIT   [CHOICE]   HELP
```

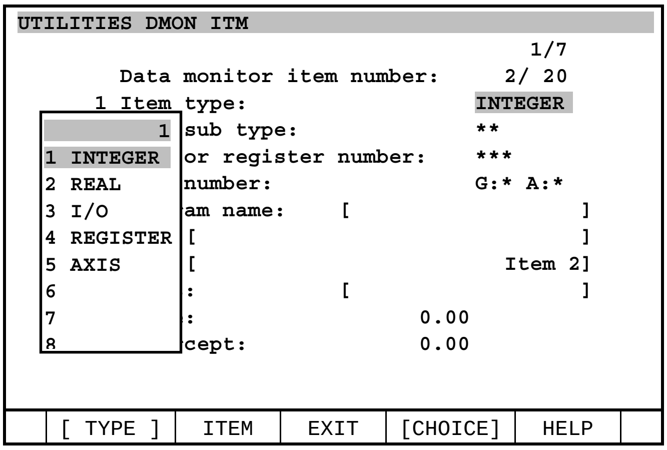
You can specify comment about the item in “4 Des” and “5 Unit”.
“6 Slope”, “7 Intercept” is for linear transformation. When Non-zero value is specified as slope, Data is transformed as follows.

Data = [Slope] * [raw data] + [intercept]

When 0 is specified as slope, raw data is used without transformation.

Notice: Transformed data is used for comparison between PAUSE/WARN limit or condition of Start/End trigger.

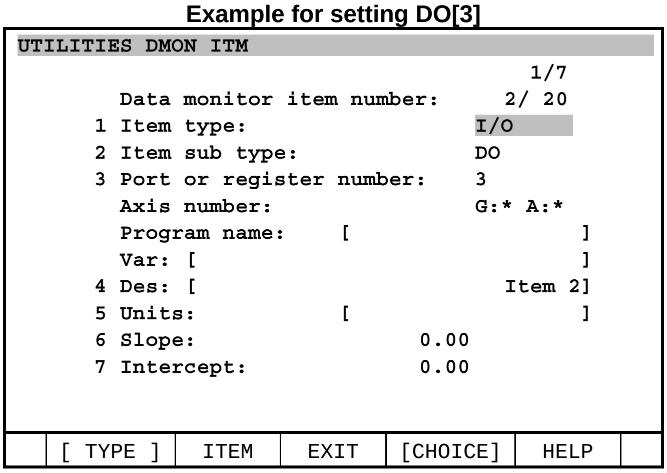
10 This is the screen you use to edit an item. Press F2, ITEM, to select a different item by number. Not all of the menu items are available for all item types. If they are not available the item is not numbered, you cannot move the cursor to it, and it displays as ***.
To set the Item type, move the cursor to line 1 and press [CHOICE]. The following screen will be displayed.



```
UTILITIES DMON ITM
                                              1/7
        Data monitor item number:           2/ 20
      1 Item type:                        INTEGER
  |          1|sub type:                  **
  |1 INTEGER  |or register number:        ***
  |2 REAL     |number:                    G:* A:*
  |3 I/O      |am name:    [                   ]
  |4 REGISTER |[                               ]
  |5 AXIS     |[                           Item 2]
  |6          |:           [                   ]
  |7          |:                    0.00
  |8          |cept:                0.00

   [ TYPE ]   ITEM    EXIT   [CHOICE]   HELP
```

11 To change the Item type, move the cursor to Item type, and press [ENTER] key.
Case that I/O is selected:
Specify “2 Item sub type” and “3 Port or register number”.

Example for setting DO[3]

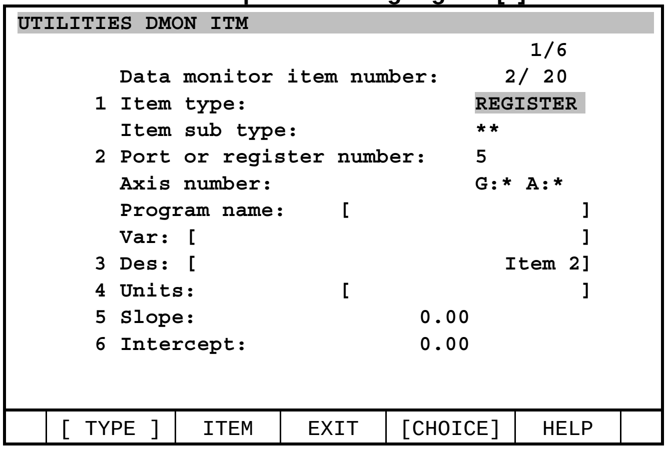


```
UTILITIES DMON ITM
                                          1/7
        Data monitor item number:       2/ 20
      1 Item type:                   I/O
      2 Item sub type:               DO
      3 Port or register number:     3
        Axis number:                 G:* A:*
        Program name:    [                  ]
        Var: [                              ]
      4 Des: [                         Item 2]
      5 Units:           [                  ]
      6 Slope:                 0.00
      7 Intercept:             0.00
```

Case that REGISTER is selected:

Specify “2 Port or register number”.

Example for setting register[5]



```
UTILITIES DMON ITM
                                          1/6
        Data monitor item number:       2/ 20
      1 Item type:                   REGISTER
        Item sub type:               **
      2 Port or register number:     5
        Axis number:                 G:* A:*
        Program name:    [                  ]
        Var: [                              ]
      3 Des: [                         Item 2]
      4 Units:           [                  ]
      5 Slope:                 0.00
      6 Intercept:             0.00
```

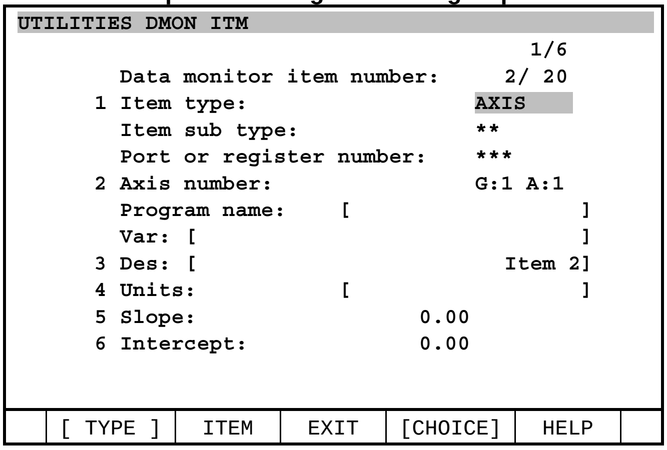
Case that Axis is selected:

Specify “2 Axis number”.

Specify group number of robot in “G:” and axis number in “A: “.

Caution: About J3 axis, J2/J3 interaction is not considered.

Example for setting J1 axis of group 1 robot



```
UTILITIES DMON ITM
                                          1/6
        Data monitor item number:       2/ 20
      1 Item type:                   AXIS
        Item sub type:               **
        Port or register number:     ***
      2 Axis number:                 G:1 A:1
        Program name:    [                  ]
        Var: [                              ]
      3 Des: [                         Item 2]
      4 Units:           [                  ]
      5 Slope:                 0.00
      6 Intercept:             0.00
```

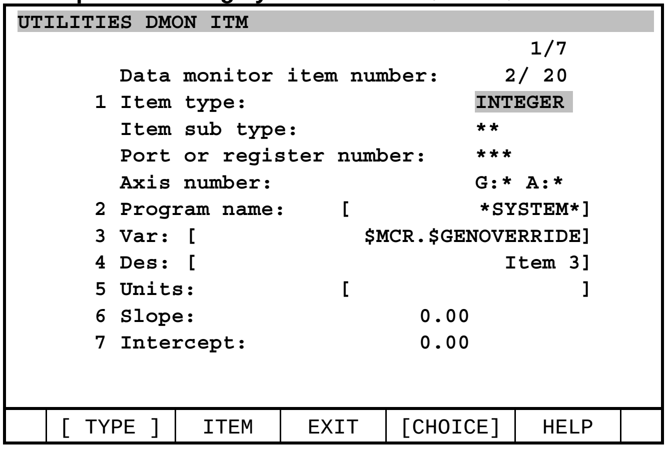
Case that INTEGER or REAL is selected:

Select Integer or Real depending on data type of system variable or KAREL variable.

Specify “2 Program name”.
Specify [*system*] for system variable or [name of KAREL program] for KAREL variable.

Specify “3 Var”.
Specify variable name in “3 Var”.

Example for setting system variable $MCR.$GENOVERRIDE



```
UTILITIES DMON ITM
                                              1/7
        Data monitor item number:     2/ 20
      1 Item type:                   INTEGER
        Item sub type:               **
        Port or register number:     ***
        Axis number:                 G:* A:*
      2 Program name:    [           *SYSTEM*]
      3 Var: [             $MCR.$GENOVERRIDE]
      4 Des: [                         Item 3]
      5 Units:           [                   ]
      6 Slope:                   0.00
      7 Intercept:               0.00

   [ TYPE ]    ITEM     EXIT    [CHOICE]   HELP
```

12 When you are done editing this item you can press F2, ITEM to select a different item by number, or press F3, EXIT to return to the setup screen.

21.2 DATA MONITOR SCHEDULE

You choose the items you want to monitor in a Data Monitor schedule.
You can also specify:

- Report file naming details
- Sampling frequencies
- WARN and STOP limits
- Start and Stop triggers

Reports

Reports are created automatically when you set the Reporting item on the Data Monitor schedule screen to ENABLED. You can specify a file name and a file device on the Data Monitor Schedule screen for the Data Monitor report.

Table 21.2 Data monitor schedule menu items

ITEM	DESCRIPTION
Schedule Comment	You can add a comment to each Data Monitor schedule.
File device	This item allows you to specify the name of the device to be used when writing a report. You can choose from FLPY:, PRN:, FR:, MC:, CONS:, or RD:.
File name Default: Blank	This item allows you to specify the name of the file to be used for a report. A .DT file extension is always used. If you leave this item blank, and Reporting is enabled, the saved data file will be named "SAMPL".
File name index Default: 0 Min: 0 Max: 999	This item allows you to specify an index number to be appended to the file name when a report is generated. If this item is non zero, each time a report is generated this index will be incremented. For example if the file name is SAMPL then successive reports will be named SAMPL001.DT, SAMPL002.DT, and so forth.

ITEM	DESCRIPTION
File size Default: 0 Min:0 Max:99999	This item specifies the amount of memory in KB you expect to use on the file device. During execution of Sample Start[n], the device is checked for this amount of free memory. If it is not available, an error is posted. If you specify 0 as the file size, the system only checks that there is at least one available block on the media.
Sampling	This item specifies the sampling frequency. -Request - This is the sampling frequency you specify. -Actual - This is the actual sampling frequency that Data Monitor will use. Since there are only certain frequencies available, it may be greater than or less than the sampling frequency you specify.
Monitoring	This item specifies the monitoring frequency. -Request - This is the monitoring frequency you specify. -Actual - This is the actual monitoring frequency that Data Monitor will use. Since there are only certain frequencies available, it may be greater than or less than the monitoring frequency you specify.
Recording	This item specifies the recording frequency. -Request - This is the recording frequency you specify. -Actual - This is the actual recording frequency that Data Monitor will use. Since there are only certain frequencies available, it may be greater than or less than the recording frequency you specify.
Record mode Default: CONTINUOUS	This item specifies the recording mode. -ONE BUFFER - Data will be recorded until the record data buffer is full. -CONTINUOUS - The record buffer is re-used when full.
Number of items Default: 1 Min: 1 Max: 6	This item specifies how many items are monitored by this particular schedule.
Start item	This item specifies the condition for starting data recording and enable/disable of this condition after Sample Start instruction. If this condition is set to disable, data recording is started just after Sample Start instruction. The condition is set as the comparison between the value of the monitoring item and setting item. For example, if condition is [recording is started after current command (item No.=2) becomes more than 150.], please set to [2 > 150]. NOTE: The user can check the item No. for monitoring by data monitor schedule screen.
Stop item	This item specifies the condition for finishing data recording and enable/disable of this condition after Sample Start instruction. If this condition is set to disable, data recording is not finished till performing Sample End instruction. The method of setting condition is the same as Start item.

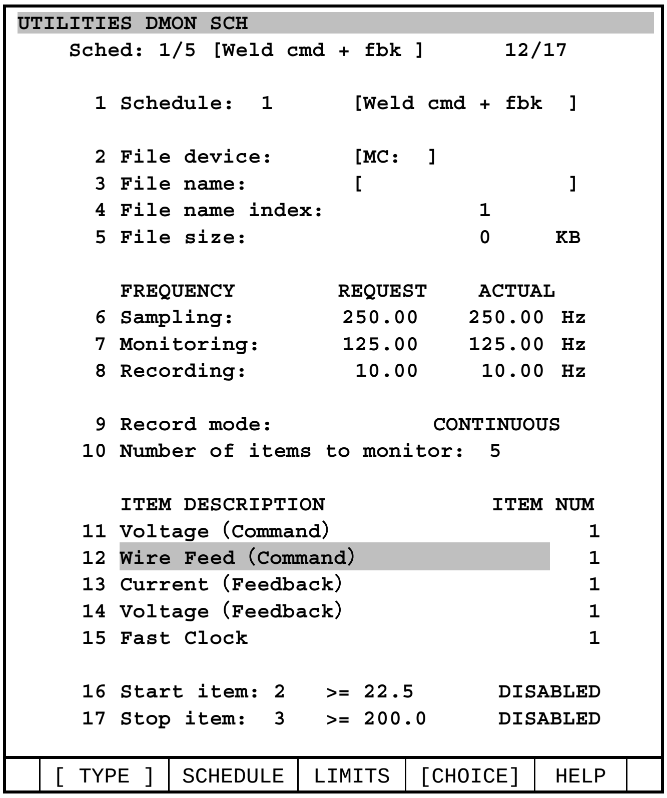
Procedure 21-2 Setting Up and Editing a Data Monitor Schedule

Condition

You have installed the Data Monitor schedule on your controller.

Step

1 Press [MENU] key.
2 Select UTILITIES.
3 Press F1, [TYPE].
4 Select Data Monitor Schedules. You will see a screen similar to the following.



```
UTILITIES DMON SCH
    Sched: 1/5 [Weld cmd + fbk ]      12/17

      1 Schedule:  1      [Weld cmd + fbk  ]

      2 File device:      [MC:  ]
      3 File name:        [               ]
      4 File name index:           1
      5 File size:                 0     KB

        FREQUENCY        REQUEST    ACTUAL
      6 Sampling:         250.00    250.00  Hz
      7 Monitoring:       125.00    125.00  Hz
      8 Recording:         10.00     10.00  Hz

      9 Record mode:             CONTINUOUS
     10 Number of items to monitor:  5

        ITEM DESCRIPTION             ITEM NUM
     11 Voltage (Command)                   1
     12 Wire Feed (Command)                 1
     13 Current (Feedback)                  1
     14 Voltage (Feedback)                  1
     15 Fast Clock                          1

     16 Start item: 2   >= 22.5       DISABLED
     17 Stop item:  3   >= 200.0      DISABLED
```

5 There are 5 Data Monitor schedules. The top line of the Schedule screen displays the current schedule number and its comment. To select a different schedule, press F2, SCHEDULE, and enter the number of the schedule you want to modify after the prompt.

6 To enter or modify the schedule comment, move the cursor to menu item 1 and press ENTER.

7 To specify the File device, move the cursor to line 2 and press F4, [CHOICE].

8 To specify the data monitor report file name, move the cursor to menu item 3 and press ENTER.

9 If you want to generate multiple report files with an index number in the file name, specify the starting index number on line 4. If you don't want to create a new indexed file each time this schedule is used, enter 0 on line 4.

10 Move the cursor to line 5, and specify the maximum report size, if desired, or leave it set to 0.

11 You specify the Sampling, Monitoring, and Recording frequencies on lines 6, 7 and 8 of the Data Monitor Schedule screen. There are two values shown for these three items, the Requested Frequency that you specify, and the Actual Frequency that will be used as the Sampling, Monitoring, or Recording Frequency. When you enter the desired frequency in the Requested column, the Actual column will update with the closest available frequency.

NOTE

The Monitoring and Recording Frequencies must be fractions of the Actual Sampling frequency. If you have an actual sampling frequency of 125 Hz, the maximum Monitoring and Recording frequency can only be 125 Hz. If you modify the Sampling frequency, the Actual frequency may change for all three frequencies.

12 To modify the Record mode, move the cursor to line 9 on the Data Monitor Schedule screen and press F4, [CHOICE].

13 Each Data Monitor schedule can monitor up to 6 items simultaneously. You can specify the number of items to monitor on line 10 of the Data Monitor Schedule screen.

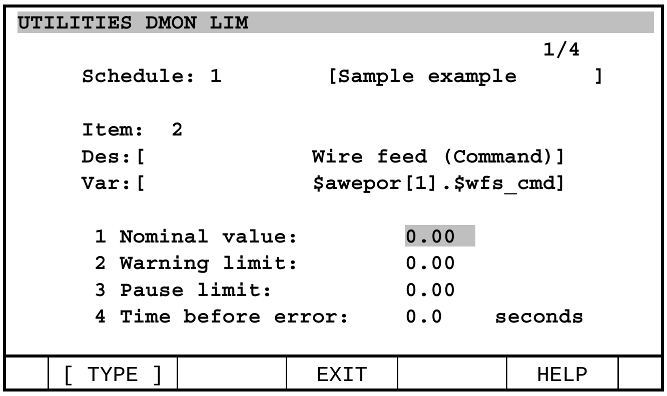
14 Specify the items you want to monitor on lines 11 through 15 of the Data Monitor Schedule screen. Press F4, [CHOICE], and select the item you want to monitor from the list of items displayed. You can also specify an item by number in the Item Num column.

Specify WARN and STOP Limits for Process Signals

15 To specify WARN and STOP limits, move the cursor to each item you have specified on lines 11 to 15 and press the F3, LIMITS function key.

16 Move the cursor to one of monitored items.

17 Press F3, LIMITS. You will see a that contains details on how specifically to monitor the item you selected.



```
UTILITIES DMON LIM
                                        1/4
     Schedule: 1        [Sample example      ]

     Item:  2
     Des:[             Wire feed (Command)]
     Var:[             $awepor[1].$wfs_cmd]

      1 Nominal value:        0.00
      2 Warning limit:        0.00
      3 Pause limit:          0.00
      4 Time before error:    0.0    seconds
```

18 Select each item and set it as desired.

19 When you finished setting items, press F3, EXIT, to display the previous screen.

Start and Stop Trigger Items

20 You can specify the Start and Stop trigger items and conditions using the fields in menu items 16 and 17 on the Data Monitor Schedule screen.

21.3 PROGRAMMING

You can use the following teach pendant instructions to start and end data monitoring:

- Sample Start[]
- Sample End

The Start instruction has a schedule number as an input parameter.

> **NOTE**
> You cannot start multiple data monitoring sessions at one time. You must end a monitoring session with a Sample End before executing another Sample Start.

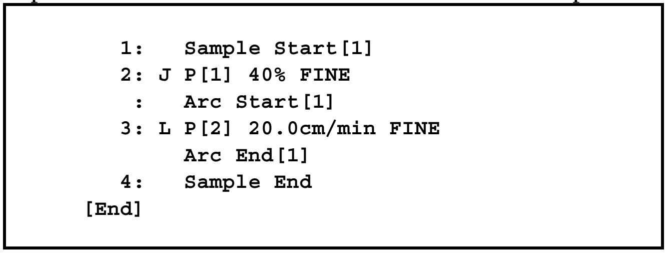
Sample Start[1]

- To start data monitoring, include the Sample Start[] instruction in a teach pendant program.

Sample End

- To stop data monitoring, include the Sample End instruction in a teach pendant program.

See Fig. 21.3 for an example of how to use these instructions in a teach pendant program.



```
   1:   Sample Start[1]
   2: J P[1] 40% FINE
    :   Arc Start[1]
   3: L P[2] 20.0cm/min FINE
        Arc End[1]
   4:   Sample End
 [End]
```

Fig. 21.3 Example of using sample start[] and sample end in a teach pendant program

The Sample Start and Sample End instructions are located in the Data Monitor category of the Teach Pendant Editor INST menu.

21.4 DATA MONITOR CHART

Data monitor chart screen enables you to display the graphs of the monitor reports on the teach pendant. The graphs of the monitor reports are displayed after reading from other memory (MC etc.). So it is needed to insert other memory to a robot controller before displaying data monitor chart screen.

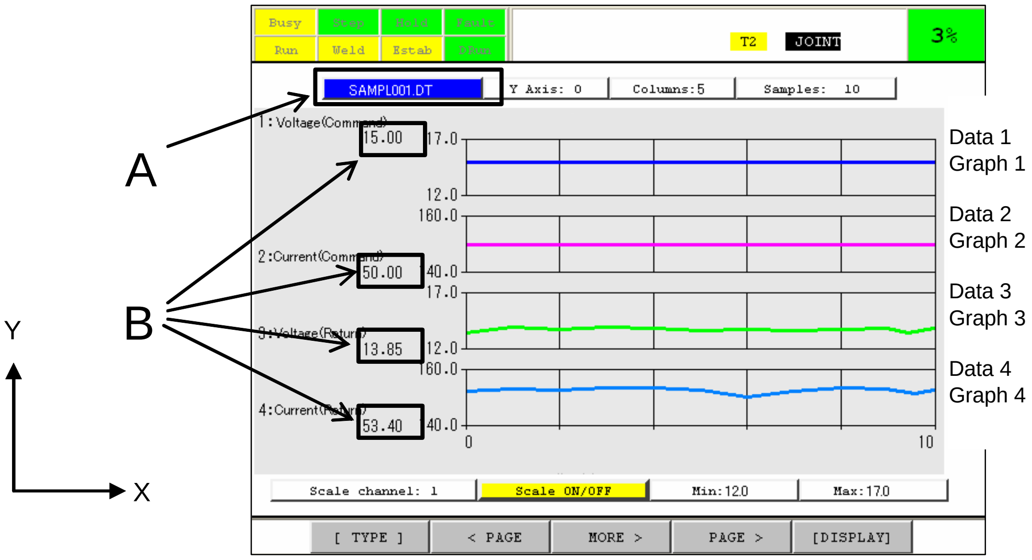
Procedure 21-3 Data monitor chart

Condition

Data monitor function is ordered.
Robot controller have already detect other memory which have monitor reports.

Step

1 Press [MENU] key.
2 Select “1 Utility”.
3 Press F1, [TYPE].
4 Select “Data Mon Chart”. The following screen will be displayed.



SAMPL001.DT
Y Axis: 0
Columns:5
Samples: 10
1:Voltage(Command) 15.00
2:Current(Command) 50.00
3:Voltage(Return) 13.85
4:Current(Return) 53.40
Scale channel: 1
Scale ON/OFF
Min:12.0
Max:17.0

Fig. 21.4 Data monitor chart screen

5 Data monitor chart screen can have up to 8 graphs. Y direction of a graph is reported data and X direction is the reported count.
6 The file name of the monitor report is displayed on Fig 21.4 A. When the cursor is moved to A and [ENTER] key is pressed, file select screen will be displayed and the user can select the “.DT” file in other memory.
7 In Fig 21.4 B, newer value of the Y direction of each graph is displayed.
8 “Y Axis:” can compare displayed graphs. If 1 is set to “Y Axis:”, Data 2-4 are displayed on Graph 1. If 2 is set to “Y Axis:”, Data 1,3,4 are displayed on Graph 2.
9 “Columns:” can specify the number of the scale of X direction. “Columns:” can be set from 1 to 20.
10 “Samples:” can specify the number of displayed data. “Samples:” can be set from 1 to 100. If the user want to the part of the data, please set small number to this item.
11 “Scale channel:”, “Scale ON/OFF”, “Min:” and “Max:” can change the scale of Y direction in each graph. First, enter the graph number which you want to change scale to “Scale channel:”. If 1 is set to “Scale channel:”, the user can change the scale of graph 1. Next, set the minimum and maximum value

of Y axis to “Min:” and “Max:”. After that, move the cursor to “Scale ON/OFF” and press [ENTER] key. Y axis scale of the graph which is specified by “Scale channel:” is changed.

12 F2, < PAGE and F4, PAGE > key can change the scale of X direction of each graph.

13 F4, PAGE > key can move the column of X axis by +1.

14 F5, [DISPLAY] key can switch display enable/disable of each graph.

22 BRAKE CHECK FUNCTION

Brake check function (A05B-2500-J951) is a preventive maintenance function. It is used to check that the motor does not move when motor torque is applied while brake is engaged. This allows user to perform early detection of potential brake failure. User is informed of the checking result by warning message or system variables.

22.1 INITIAL SETTING BEFORE USE

Enable/Disable switch for brake check function

The following system variable is the switch that enable/disable brake check function.

`$BCCFG.$BCK_ENABLE` TRUE: Enable (default) / FALSE: Disable

In order to check or modify the system variables, press [MENU] key and select “SYSTEM”. Then, press F1, [TYPE] key and select “Variables”. The system variables screen will be displayed.

Please confirm this variable is TRUE (Enable) before use. The default value of this variable is TRUE (Enable). If this variable is FALSE (Disable), brake check never starts.

Enable/Disable switch for each axis

On this function, the user can customize enable/disable brake check for each axis. Please set the following system variable to TRUE (Enable) for the axis where you want to execute brake check.

`$BCK_GRP[group].$CHK_AXIS[axis]` TRUE: Enable / FALSE: Disable

On robot axis, the default value is TRUE (Enable). On except robot axis (aux axis, servo gun, or etc.), the default value is FALSE (Disable).

In addition, if you execute brake check on dual drive axis, please set this variable to TRUE (Enable) for both master and slave axis.

Setting brake torque

This function needs brake torque for each axis in order to determine torque command during brake check. Please set the following system variable with the brake torque [kgf*cm] for each axis.

`$BCK_GRP[group].$BRK_TORQUE[axis]` brake torque [kgf*cm]

On robot axis, this variable is set automatically, however, on except robot axis (aux axis, servo gun, or etc.), this variable is not set automatically, so please manually set this variable with brake torque if you execute brake check on except robot axis. For specification value of brake torque, please refer to “BRAKE SPECIFICATIONS” on AC SERVO MOTOR DESCRIPTIONS corresponding to your motor. Please note that the unit of this variable is [kgf*cm].

AC SERVO MOTOR DESCRIPTIONS	Specification number
AC SERVO MOTOR α series DESCRIPTIONS	B-65142EN
AC SERVO MOTOR β series DESCRIPTIONS	B-65232EN
AC SERVO MOTOR α*i* series DESCRIPTIONS	B-65262EN
AC SERVO MOTOR β*i* series DESCRIPTIONS	B-65302EN

In addition, if you execute brake check on dual drive axis (except robot axis), please set this variable as follows. (For robot axis, it is no need to set even if it's dual drive axis.)

Case-1: Either master or slave motor has one brake.
Please set this variable with the brake torque [kgf*cm] for only the axis that has brake. For the axis that has no brake, please set 0.0 (default).

Case-2: Both master and slave motors have brake respectively.
Please set this variable with the brake torque [kgf*cm] for both master and slave axes. And then the brake torque value of master and slave axis must be same. If these values are different in the Case-2, it is not supported.

22.2 START BRAKE CHECK

To start brake check, please set the following system variable to TRUE.

$BCCFG.$BCK_START TRUE: Brake check is running. / FALSE: Brake check is finished.

User can start brake check by parameter instruction of TP program or manually setting on system variable screen. This variable automatically returns to FALSE when brake check is finished. Please not to move robot till this variable automatically returns to FALSE. In addition, if brake check function is disabled (**$BCCFG.$BCK_ENABLE=FALSE**), this variable immediately returns to FALSE even if this variable is set to TRUE.

Follows is a sample of TP program that execute brake check.

```
1: J P[1] 100% FINE
2: L P[2] 2000mm/sec FINE
3:  WAIT 1.00 (sec)
4:  $BCCFG.$BCK_START=1
5:  WAIT $BCCFG.$BCK_START=0
6: L P[3] 2000mm/sec FINE
```

Brake check cannot be started while the robot is moving as specification, so this program waits for robot to stop completely by WAIT instruction at Line 3. And this program starts brake check at Line 4 and waits to finish brake check at Line 5. As just described on this sample program, brake check must be executed under the robot stops completely.

> ⚠ **NOTE**
> In case of brake failure, the motor can move during brake check. So please be sure to execute brake check in sufficiently spaced layout. Total motion command per single check is vary by robot or axis, however, it is up to 200000[pulse]. This is about 0.4[rev] in motor revolution and about 1.4[deg] in degree of the angle assuming the gear ratio as 1/100.

Additionally, in case where the power is shut OFF during brake check for some reason, the following alarm will be posted at next power ON. This alarm indicates abnormal termination at last check.

```
BRCH-003  Last check was failed
```

The following system variable will be also set to TRUE at next power ON.

$BCCFG.$LAST_FAILED TRUE: Last check was failed. / FALSE: Last check was normally finished.

In this case, the results of brake check will not be generated, so please execute brake check again at next power ON. If next brake check would be normally finished, this variable will return to FALSE automatically.

22.3 RESULTS OF BRAKE CHECK

$BCCFG.$BCK_START automatically returns to FALSE when brake check finishes. And then the results are also posted as warning message.
If all brake passes as “Good” brake, the following message will be posted.

```
BRCH-001  All brake are normal
```

By contrast, if brake failure is detected as “Bad” brake, the following message will be posted.

Ex) Brake failure is detected on group1/J2.

```
BRCH-002  Brake abnormal (G:1,A:2)
```

The results are also recorded on the following system variable at the same time. The value of this variable is kept till next brake check starts.

$BCK_GRP[group].$CHK_RESULT[axis] TRUE: Brake failure/FALSE: Normal brake (default)

22.4 LIMITATIONS

Brake check function cannot be executed under the following conditions, or alarm will be posted.

- Calibration has not been done.
- Robot is moving.
- Servo is OFF.
- Servo parameter is under updating.
- Dual motor is under the adjusting mode.
- Even though the axis has brake and **$BCK_GRP[group].$CHK_AXIS[axis]** is TRUE, brake torque (**$BCK_GRP[group].$BRK_TORQUE[axis]**) is not set.
- Teach Pendant is enabled.
- Any axis is near by stroke limit when brake check starts.

22.5 CAUTION

- It takes about 10[sec] for single brake check execution.
- When the motor is heating up, the motor torque can decrease, so it may not be able to detect brake failure. To detect it with high accuracy, please execute brake check preferably when the motor is in low-temperature, e.g. first thing in the morning.

23 PANEL WIZARD

The Panel Wizard option allows you to use the controller to create up to four operation panels to be displayed on the teach pendant. You don’t need PC to create panel.

To use this function, Panel wizard option (A05B-2600-R594) is required.

NOTE

Panel wizard is included in KAREL use support (A05B-2600-J971) in 7DC1 and 7DD0 series. However, KAREL use support is divided to Customize support (A05B-2600-J742) and Panel wizard from 7DC2 series. If the same function of KAREL use support is needed, please order both Customize support and Panel wizard.

NOTE

Please use the teach pendant which equips the touch panel for this function.

23.1 SETTING UP

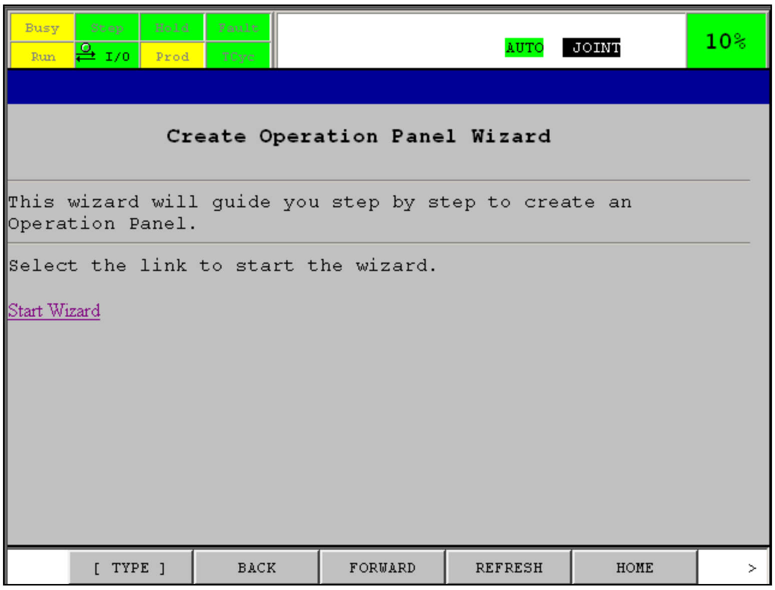
23.1.1 Overview

Use procedure 23-1 to setup the Panel Wizard.

Procedure 23-1 Setting up the Panel Wizard

Step

1 Press [MENU] key.
2 Select BROWSER.
3 Press F1, [TYPE].
4 Select Panel Wizard. The following screen will be displayed.



Create Operation Panel Wizard

This wizard will guide you step by step to create an Operation Panel.

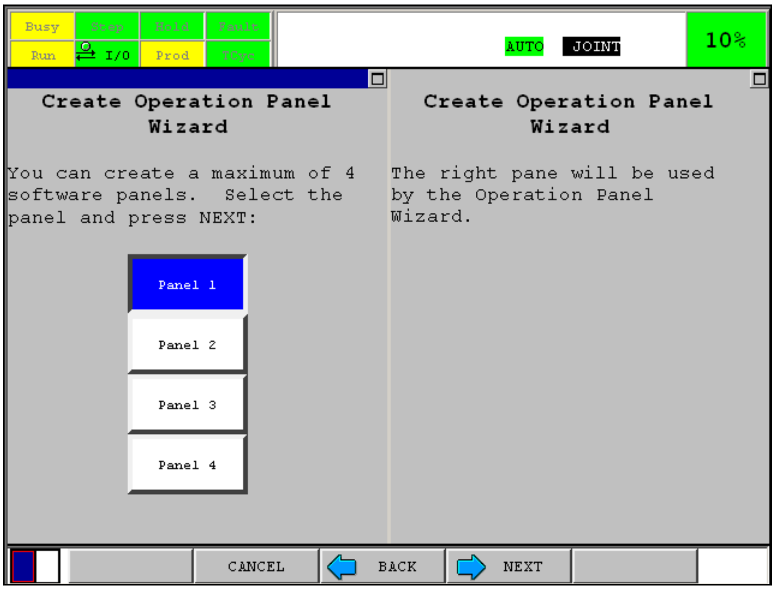
Select the link to start the wizard.

Start Wizard

Fig. 23.1.1 (a) Panel wizard start screen

5 Select Start Wizard. The following screen will be displayed.

> **NOTE**
> Double Pane mode will be displayed automatically.



Create Operation Panel Wizard

You can create a maximum of 4 software panels. Select the panel and press NEXT:

Create Operation Panel Wizard

The right pane will be used by the Operation Panel Wizard.

Fig. 23.1.1 (b) Panel selection screen

> **NOTE**
> After you make a selection on any screen, press NEXT. To cancel the Panel Wizard at any time, press F2, CANCEL. To display the previous screen at any time, press F3, BACK.

6 Select the panel that you want to create. Refer to Fig. 23.1.1 (c) for a flowchart overview of the steps required to create an operator panel.

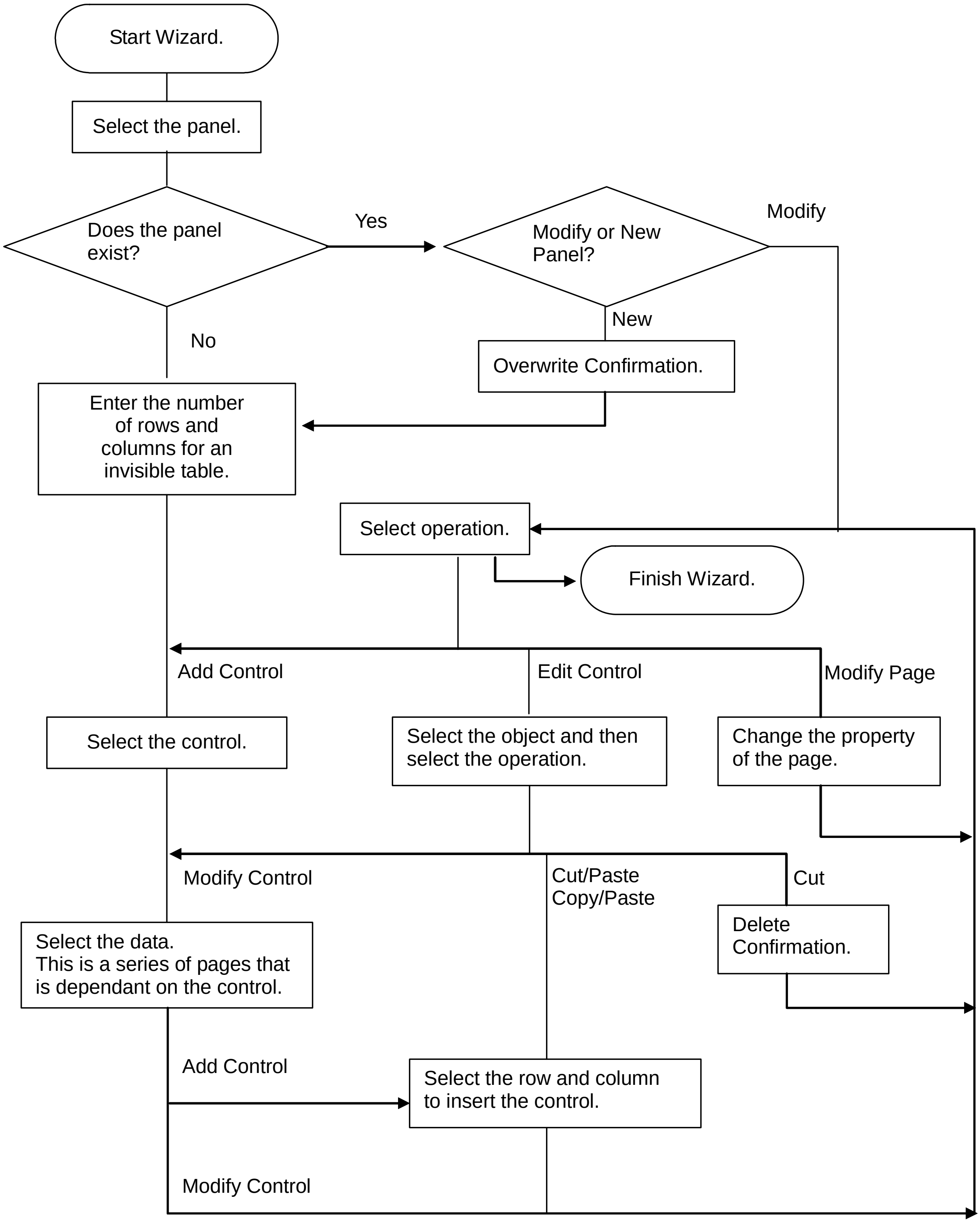


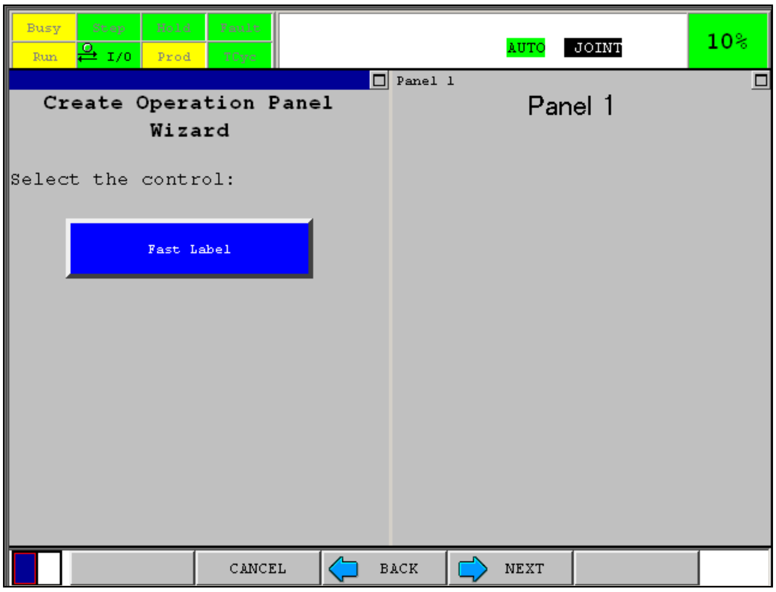
Fig. 23.1.1 (c) Create operation panel flowchart

NOTE

Control in the chart is *i*Pendant control. It corresponds to parts like buttons displayed by Panel Wizard.

23.1.2 Available *i*Pendant Controls

Panel wizard create operation panel by placing *i*Pendant control on invisible table. *i*Pendant Control to add is selected in a screen similar to following. Please refer to Fig. 23.1.1 (c) for how to display the screen.



Create Operation Panel Wizard

Select the control:

Fig. 23.1.2 *i*Pendant control selection screen

Pressing button on the center of the screen displays lists of available *i*Pendant Controls Following table lists the choices.

Table 23.1.2 Available *i*Pendant controls

Choice	Description
Fast Label	This is simply shorted version of another choice “Label”. It is described later. Refer to the description.
Fast lamp	This is simply shorted version of another choice “Toggle Lamp”. It is described later. Refer to the description.
Fast Switch	This is simply shorted version of another choice “Toggle Button”. It is described later. Refer to the description.
Button Change	This control is used to display the specified web page. It can also be used to run KAREL program.
Command Button	This control is used to write the specified value to a Register, System or KAREL Variable (except XYZWPR type) or I/O whenever the button is pushed.
EditBox	This control is used to change the value of a Register, System or KAREL Variable (except XYZWPR type) or I/O.
Label	The value of a Register, System or KAREL Variable or I/O is displayed.
Toggle Button	This control is used to change the value of a Register, System or KAREL Variable (except XYZWPR type) or I/O to the specified value following the ON (Pushed)/OFF (Popped) status of the button.
Toggle Lamp	This control is used to change the color of the control if the value of a Register, System or KAREL Variable (except XYZWPR type) or I/O fulfills the specified condition with the specified value or not.

Setup of each control is described following sections.

NOTE

Wizard is always display in double pane mode. Setup is done in the left pane. Right pane is used to display rough image of result of setup. Because display on the right pane depends on content of setup, figures in this manual usually do not match what you see during setup.

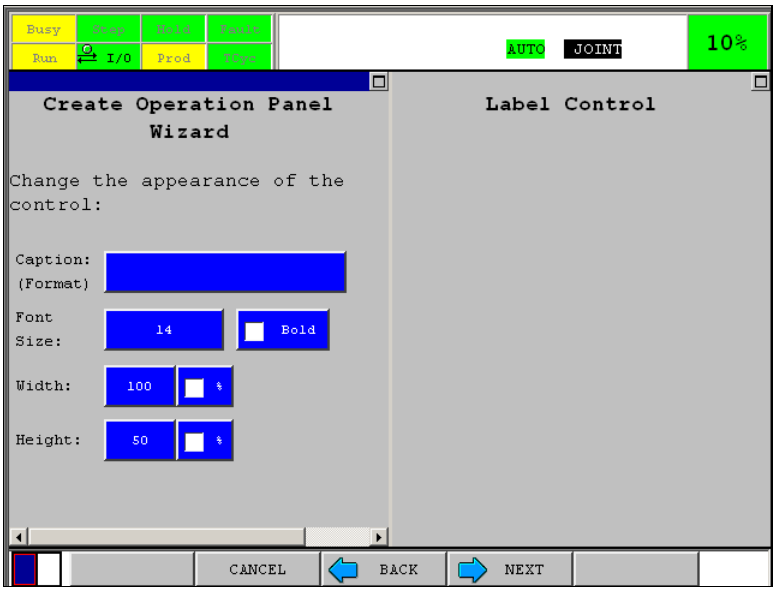
23.1.3 Setting up Fast Label

Fast lamp is shorted version of Label control. Label control displays value of a Register, System or KAREL Variable or I/O. Use procedure 23-2 to add Fast Label.

Procedure 23-2 Addition of Label control by Fast Label

Step

1 Select Fast Label in *i*Pendant Control selection screen (Fig. 23.1.2).
2 Press F4. The following screen will be displayed.



Create Operation Panel Wizard

Change the appearance of the control:

Caption: (Format)

Font Size: 14 □ Bold

Width: 100 □ %

Height: 50 □ %

Label Control

Fig. 23.1.3 (a) Size setup screen

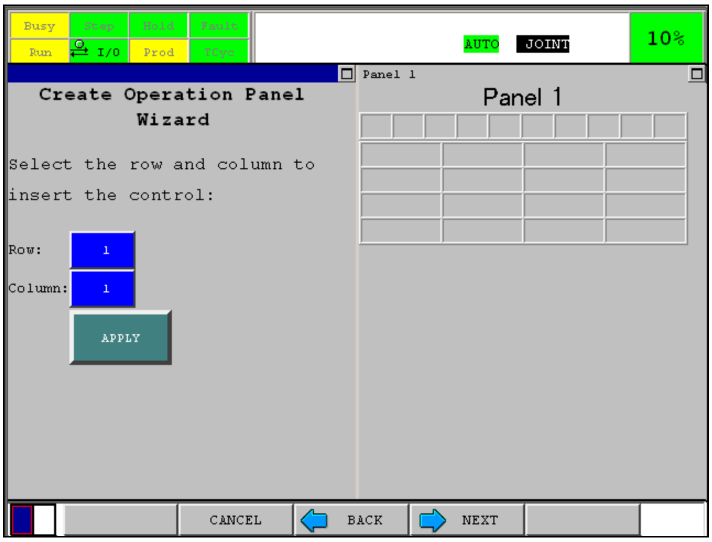
Table 23.1.3 (a) Items in size setup screen

Item	Description
Caption (format)	If data type is static (this is always true for Fast Label), input string is text displayed. If the other data type is used, caption is format string.
Font Size	Size of font. Choices are 14,16,18 and 24.
Bold	If this is checked, value is displayed in bold.
Width and %	This is width of control. If % is not checked, width is specified in pixels. If % is checked, width is specified in percentage of screen size.
Height and %	This is height of control. Unit is specified in the same way as width.

NOTE

In setup of Fast Label, data type is default to static. Input string to be displayed to Caption.

3 Setup items as required.
4 Press F4, NEXT. The following screen will be displayed.



Create Operation Panel Wizard

Select the row and column to insert the control:

Row: 1

Column: 1

Panel 1

Fig. 23.1.3 (b) Control insertion screen

Table 23.1.3 (b) Items in control insertion screen

Item	Description
Row and Column	Row and column to insert *i*Pendant Control
APPLY	This is not setup item. Pressing the button displays rough image of screen after insertion is displayed on right pane. Please note that screen image in right pane is just a rough guide.

5 Press F4, NEXT. This is end of addition of Label control by Fast Label.

23.1.4 Setting up Fast Lamp

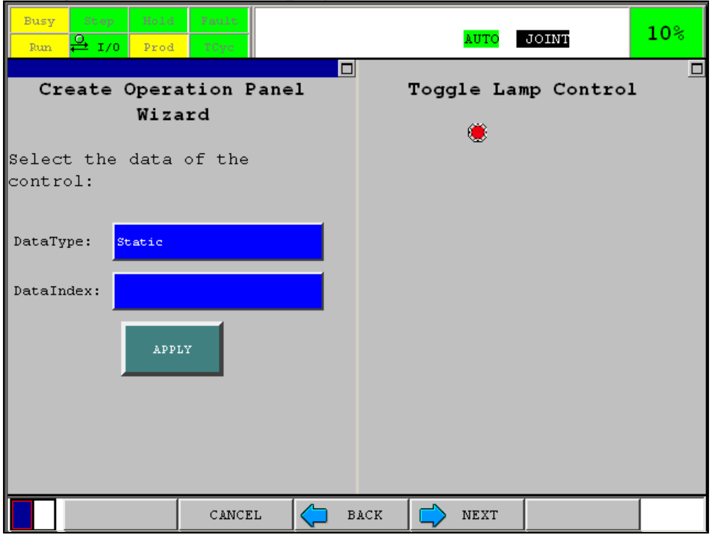
Fast lamp is shorted version of Toggle lamp. Toggle lamp is used to change the color of the control if the value of a Register, System or KAREL Variable (except XYZWPR type) or I/O fulfills the specified condition with the specified value or not. Use procedure 23-3 to add Fast Lamp.

Procedure 23-3 Addition of toggle lamp by Fast Lamp

Step

1 Select Fast Lamp in *i*Pendant Control selection screen (Fig. 23.3.2).

2 Press F4. The following screen will be displayed.



Create Operation Panel Wizard

Select the data of the control:

DataType: Static

DataIndex:

Toggle Lamp Control

Fig. 23.1.4 (a) Date type setup screen

DataType specifies the type of the data to be monitored or modified depending on the type of control. DataIndex specifies the number or the variable name associated with the DataType.

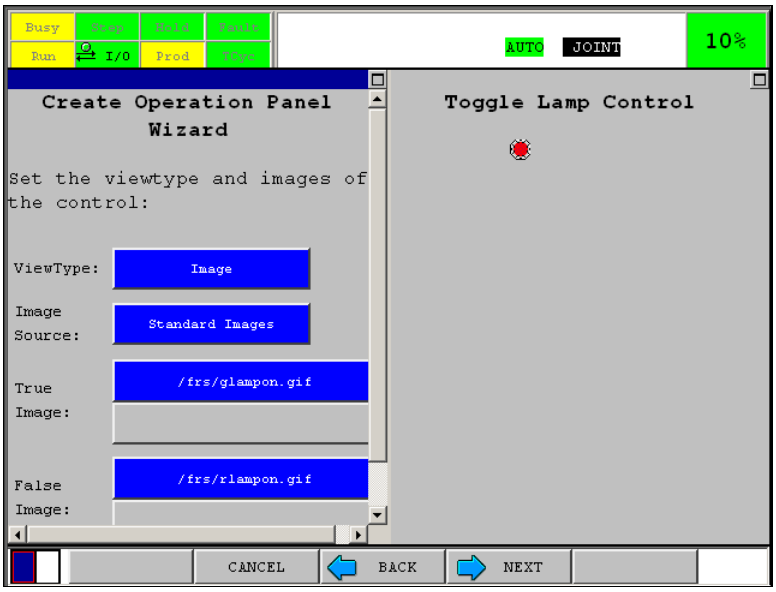
Table 23.1.4 (a) Data type and data index

Data type	Description
Static	String in caption is displayed.
Numeric Register	The value of numeric register specified in Date index is used.
System variable	The value of system variable specified in Data index is used. For System Variables, the type must be Integer, Real, Boolean, Short, Byte, or String. For example, $MNUFRAMENUM[1]
KAREL variable	The value of KAREL variable specified in Data index is used. For KAREL Variables, enclose the program name inside […]. The type must be Integer, Real, Boolean, Short, Byte, or String. Following is example of variable STR_VAR of KAREL program USEREXT. [USEREXT]STR_VAR
Dictionary Element	Dictionary element specified in Data index is used. For Dictionary Elements, specify the dictionary name and enclose the dictionary element inside […]. For example, TPAR[5] Only Label control uses dictionary element.
DI	The value of DI specified in Data index is used.
DO	The value of DO specified in Data index is used.
…….. (the other I/O type)	The value of I/O type specified in Data type and Data index is used. Data index is port number.

Toggle lamp uses specified data to decide ON/OFF of lamp.

3 Setup items as required.

4 Press F4, NEXT. The following screen will be displayed.



Create Operation Panel Wizard

Set the viewtype and images of the control:

ViewType: Image

Image Source: Standard Images

True Image: /frs/glampon.gif

False Image: /frs/rlampon.gif

Toggle Lamp Control

Fig. 23.1.4 (b) Image setup screen

Table 23.1.4 (b) Items in image setup screen

Item	Description
View type	Type of button to display Panel: specify panel type. Circle: specify circle type. Fixed circle: specify circle with fixed size. Image: specify image type.
Image Source	This is active when view type is image. Standard Images: image files installed can be selected in the menu of True image and false image. fr:*.gif：GIF file in FR: can be used. FR is device on F-ROM of controller. fr:*.jpg：JPG file in FR: can be used. Keyboard Entry: Input path of image file.
True Image	This specifies the image to be displayed when the read value fulfill the condition expression. Used only in case that ViewType is Image type. When image source is Keyboard Entry, button of lower side is active. Input path of image file by the button. When image source is not Keyboard Entry, button of upper side is active. The menu is displayed by the button. Select image fie from the sub menu.
False Image	This specifies the image to be displayed when the read value does not fulfill the condition expression. This is used only in case that View Type is Image type. Refer to description of True Image for how to specify image file.

5 Setup items as required.
6 Press of F4, NEXT leads you to control insertion screen. Please refer to Subsection 23.1.3.
7 Decide position to insert and press F4, NEXT. This is end of addition of Toggle Lamp by Fast Lamp.

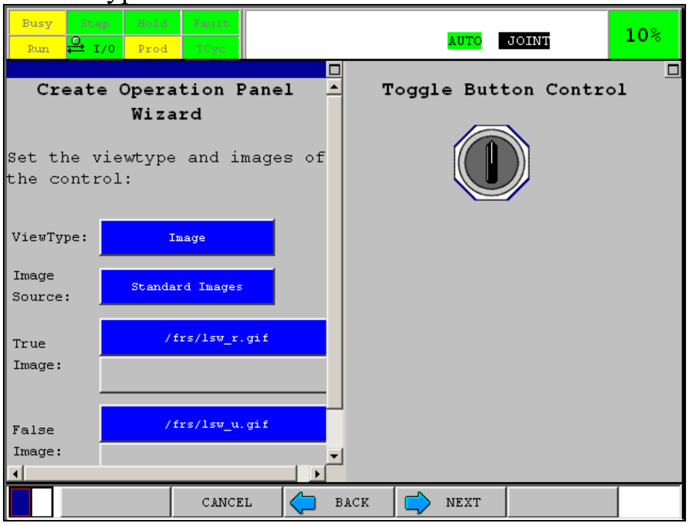
23.1.5 Setting up Fast Switch

Fast switch is shorted version of Toggle Button. Toggle button is used to change the value of a Register, System or KAREL Variable (except XYZWPR type) or I/O to the specified value following the ON (Pushed)/OFF (Popped) status of the button. Use procedure 23-4 to add Toggle button by Fast Switch.

Procedure 23-4 Addition of Toggle button by Fast Switch

Step

1 Select Fast Switch in *i*Pendant Control selection screen (Fig. 23.1.2).
2 Press of F4, NEXT leads you to data type setup screen. Please refer to Subsection 23.1.4.
3 Press of F4, NEXT leads you to image setup screen (Fig. 23.1.4 (b)). Screen is same as Subsection 23.1.4. But choices of view type are different.



Create Operation Panel Wizard

Set the viewtype and images of the control:

ViewType: Image

Image Source: Standard Images

True Image: /frs/lsw_r.gif

False Image: /frs/lsw_u.gif

Toggle Button Control

Fig. 23.1.5 Toggle button image setup screen

Table 23.1.5 Items in toggle button image setup screen

Item	Description
View type	Type of button to display Panel: specify panel type. Check box: specify check box type. Image: specify image type. F2-F5, F7-F10: specify function key type. Circle: specify circle. (Size of circle is fixed. This is just like fixed circle.)
Image Source	This is same as toggle lamp. Please refer to table 23.1.4 (b).
True Image	This specifies the image to be displayed when the button status is ON. This is used only in case that ViewType is image type. How to setup is just same as toggle lamp. Please refer to table 23.3.4 (b).
False Image	This specifies the image to be displayed when the button status is OFF. This is used only in case that ViewType is image type. How to setup is just same as toggle lamp. Please refer to table 23.3.4 (b).

4 Press of F4, NEXT leads you to control insertion screen. Please refer to Section 23.3.3.

5 Decide position to insert and press F4, NEXT. This is end of addition of Toggle button by FAST Switch.

23.1.6 Addition of Button Change Control

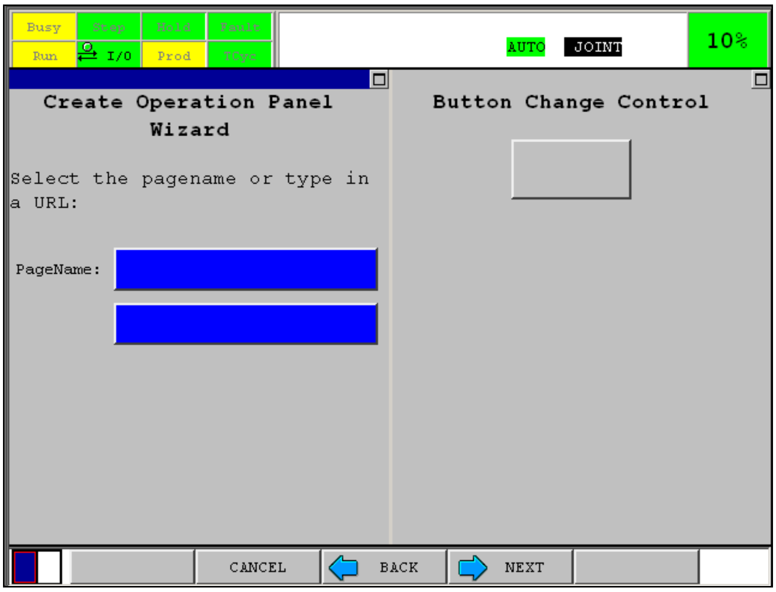
Button Change control is used to display the specified web page. It can also be used to run KAREL program. Method to run KAREL program is described later. Use Procedure 23-5 to add Button change control.

Procedure 23-5 Addition of Button Change Control

Step

1 Select Button Change in *i*Pendant Control selection screen (Fig. 23.3.2).

2 Press F4. The following screen will be displayed.



Create Operation Panel Wizard

Select the pagename or type in a URL:

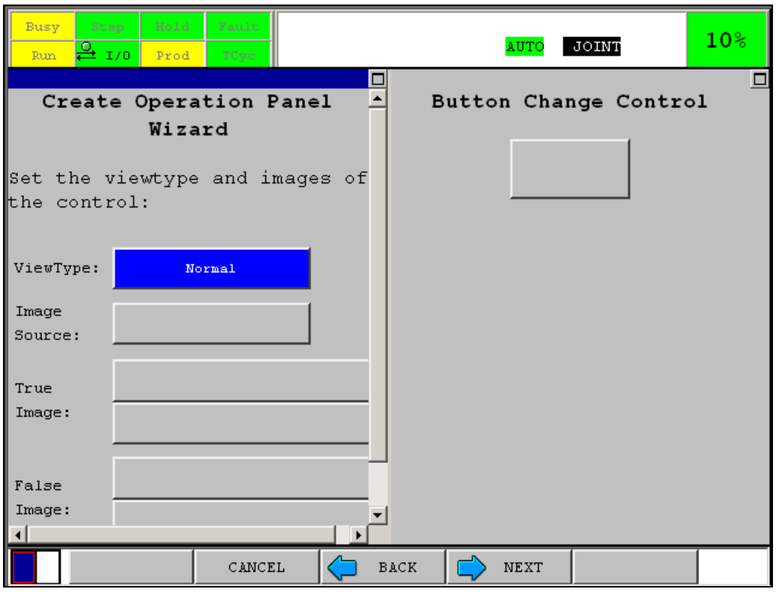
PageName:

Fig. 23.1.6 (a) URL setup screen

Table 23.1.6 (a) URL setup screen

Item	Description
PageName	Press of upper side button displays STM/HTM files in FR:. Select file to display. If you want to display the other file, press lower side button and input URL.

3 Press of F4, NEXT, leads you to Button Change image setup screen. Screen is same as image setup screen of section 23.2.4. But choices of view type are different.



Create Operation Panel Wizard

Set the viewtype and images of the control:

ViewType: Normal

Image Source:

True Image:

False Image:

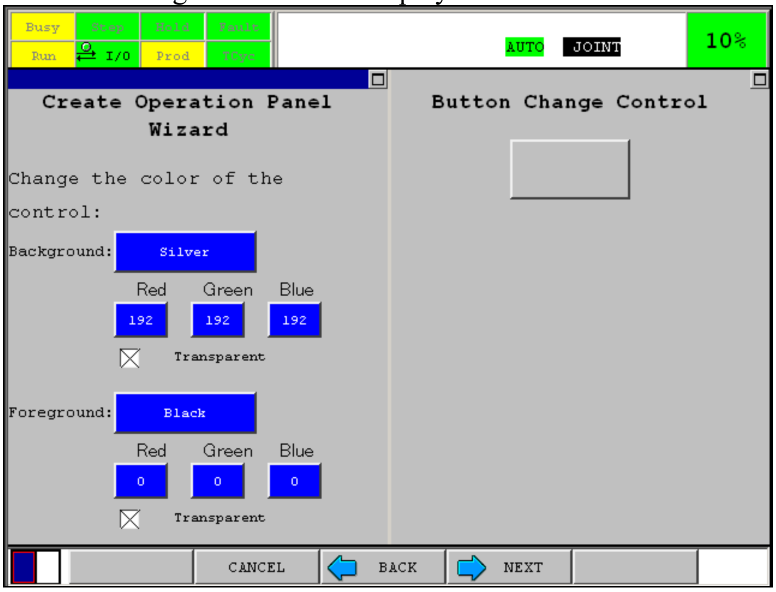
Button Change Control

Fig. 23.1.6 (b) Button change image setup screen

Table 23.1.6 (b) Items in Button change image setup screen

Item	Description
View type	Type of button to display Normal: Specify the normal button. Image: specify image type. F2-F5, F7-F10: specify function key type.
Image Source	This is same as toggle lamp. Please refer to table 23.3.4 (b).
True Image	This specifies the image to be displayed when the button is pushed. Used only in case that ViewType is Image type. How to setup is just same as toggle lamp. Please refer to table 23.3.4 (b).
False Image	Specify the image to be displayed when the button is not pushed. Used only in case that ViewType is Image type. How to setup is just same as toggle lamp. Please refer to table 23.3.4 (b).

4 Press F4, NEXT. The following screen will be displayed.



Create Operation Panel Wizard

Change the color of the control:

Background: Silver

Red 192 Green 192 Blue 192

☒ Transparent

Foreground: Black

Red 0 Green 0 Blue 0

☒ Transparent

Button Change Control

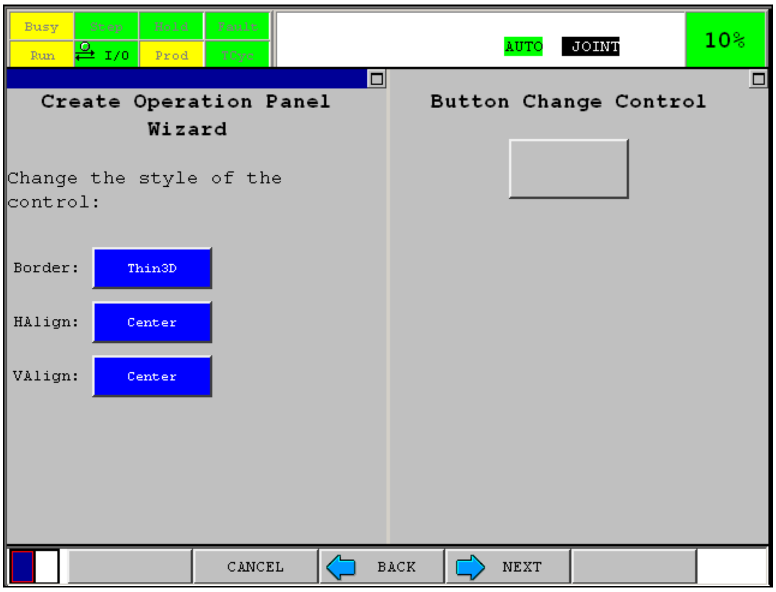
Fig. 23.1.6 (c) Color setup screen

Table 23.1.6 (c) Items in color setup screen

Item	Description
BackGround	This specifies background color. Pressing button displays list of standard colors in the menu.
(BackGround) Red, Green, Blue	Specify background color by RGB. This enables more detailed setup than selection from menu of BackGround button.
(BackGround) Transparent	With check of this item, background color of control becomes background color of web page.
ForeGround	This specifies color of character. How to setup is same as background color.
(ForeGround) Red, Green, Blue	Please refer to the same item for background color.
(ForeGround) Transparent	With check of this item, foreground color of the control becomes foreground color of web page.

5 Press of F4, NEXT leads you to size setup screen (Fig. 23.2.3 (a)). Please refer to Section 23.3.3. Caption is used as fixed string. It is not used as format.

6 Press F4, NEXT. The following screen will be displayed.



Create Operation Panel Wizard

Change the style of the control:

Border: Thin3D

HAlign: Center

VAlign: Center

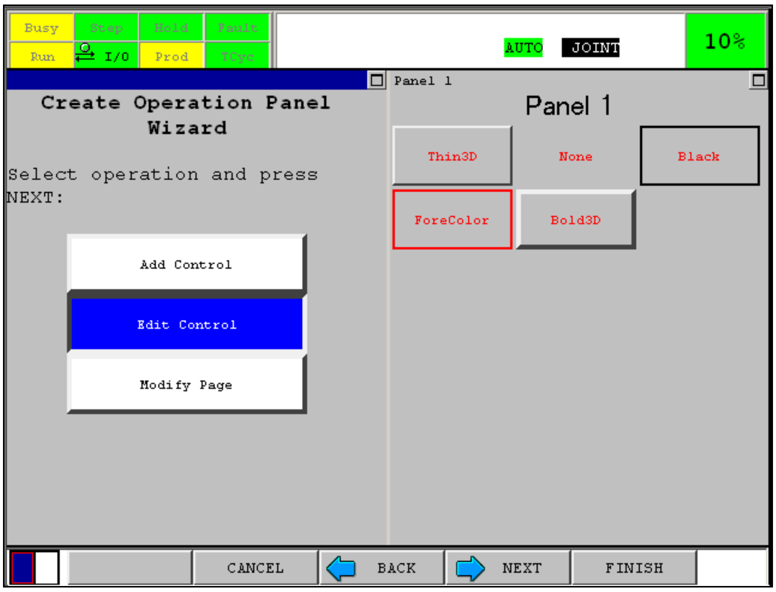
Button Change Control

Fig. 23.1.6 (d) Style setup screen

Table 23.1.6 (d) Items in style setup screen

Item	Description
Border	This specifies type of borderline of button. Choices are Thin 3D, None, Black, ForeColor and Bold 3D.
HAlign	This specifies alignment of string in horizontal direction. Choices are Left, Center and Right.
VAlign	This specifies alignment of string in vertical direction. Top, Center and Bottom.

Following figure shows difference between types of borderline. The right pane shows buttons of each 5 border types. This figure is just a sample that has nothing to do with flow to add button change control.



Create Operation Panel Wizard

Select operation and press NEXT:

Fig. 23.1.6 (e) Type of borderlines

7 Press of F4, NEXT leads you to control insertion screen. Please refer to Subsection 23.1.3.

8 Decide position to insert and press F4, NEXT. This is end of addition of Button Change control.

23.1.7 Addition of Command Button Control

Command Button Control is used to write the specified value to a Register, System or KAREL Variable (except XYZWPR type) or I/O whenever the button is pushed. Use Procedure 23-6 to add Command Button Control.

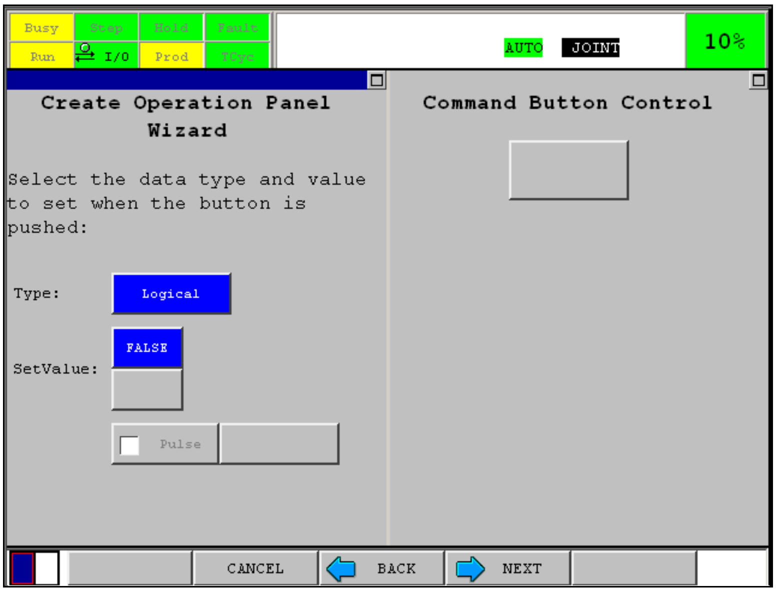
Procedure 23-6 Addition of Command Button Control

Step

1 Select Command Button in *i*Pendant Control selection screen (Fig. 23.2.2).

2 Press of F4, NEXT leads you to data type setup screen. Please refer to section 23.2.4.

3 Press F4, NEXT. You will see a screen similar to the following.



Create Operation Panel Wizard

Select the data type and value to set when the button is pushed:

Type: Logical

SetValue: FALSE

Fig. 23.1.7 Command button setup screen

Table 23.1.7 Items in command button setup screen

Item	Description
Type	Logical: Data is transacted as bool type. (SetValue is TRUE/FALSE.) Numerical: Data is transacted as numerical type. The value of SetValue is applied.
SetValue	This specifies the value written when the button is pushed.
Pulse	This is active only when data type is DO. If this is checked, signal output becomes pulsed output.
Pulse width (right item of pulse check box)	This specifies width of pulse in msec.

4 Press of F4, NEXT leads you to image setup screen. This is same as Button change image setup screen (Fig. 23.3.6 (b)). Please refer to section 23.3.6.
5 Press of F4, NEXT leads you to color setup screen. Please refer to section 23.3.6.
6 Press of F4, NEXT leads you to size setup screen. Please refer to section 23.3.3. Caption is used as fixed string. It is not used as format.
7 Press of F4, NEXT leads you to style setup screen. Please refer to section 23.3.6.
8 Press of F4, NEXT lead you to control insertion screen. Please refer to section 23.3.3.
9 Decide position to insert and press F4, NEXT. This is end of addition of Command Button Control.

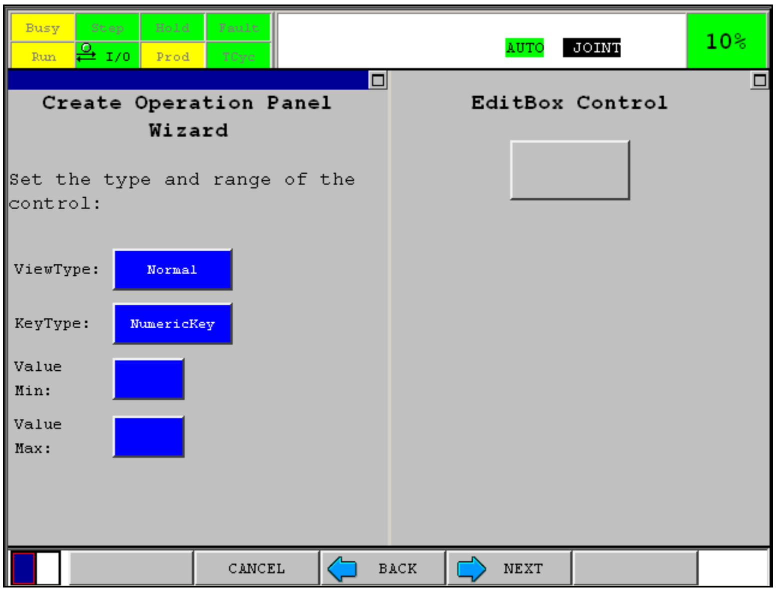
23.1.8 Addition of Edit Box Control

Edit Box Control is used to change the value of a Register, System or KAREL Variable (except XYZWPR type) or I/O. Use Procedure 23-7 to add Edit Box Control.

Procedure 23-7 Addition of Edit Box Control

Step

1 Select Edit Box in *i*Pendant Control selection screen (Fig. 23.3.2).
2 Press of F4, NEXT leads you to data type setup screen. Please refer to section 23.3.4.
3 Press F4, NEXT. You will see a screen similar to the following.



Create Operation Panel Wizard

Set the type and range of the control:

ViewType: Normal

KeyType: NumericKey

Value Min:

Value Max:

EditBox Control

Fig. 23.1.8 Edit box setup screen

Table 23.1.8 Items in edit box setup screen

Item	Description
View type	Normal: specify simple button. F2-F5, F7-F10: specify function key type.
KeyType	Select the type of virtual keyboard. NumericKey: virtual keyboard for number input FullKey: virtual keyboard for string input
Value Min, Value Max	Minimum and maximum value of numerical value.

4 Press of F4, NEXT leads you to color setup screen. Please refer to section 23.2.6.
5 Press of F4, NEXT leads you to size setup screen. Please refer to section 23.2.3.
6 Press of F4, NEXT leads you to style setup screen. Please refer to section 23.2.6.
7 Press of F4, NEXT lead you to control insertion screen. Please refer to section 23.2.3.
8 Decide position to insert and press F4, NEXT. This is end of addition of Edit Box Control.

23.1.9 Addition of Label Control

Label control displays value of a Register, System or KAREL Variable or I/O. Use procedure 23-8 to add Fast Label.

Procedure 23-8 Addition of Label Control

Step

1 Select Label in *i*Pendant Control selection screen (Fig. 23.2.2).
2 Press of F4, NEXT leads you to data type setup screen. Please refer to section 23.2.4.
3 Press of F4, NEXT leads you to size setup screen. Please refer to section 23.2.3.
4 Press of F4, NEXT leads you to style setup screen. Please refer to section 23.2.6.
5 Press of F4, NEXT leads you to color setup screen. Please refer to section 23.2.6.
6 Press of F4, NEXT lead you to control insertion screen. Please refer to section 23.2.3.
7 Decide position to insert and press F4, NEXT. This is end of addition of Label Control.

Setup of Fast Label included step 3 and 6 only. Default value is used for items in the other steps. You can add Label Control by Fast Label and then modify it.

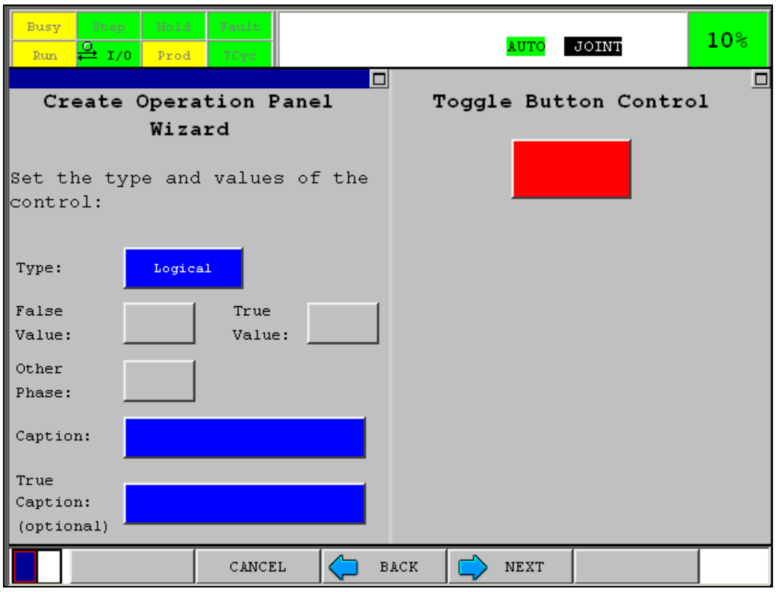
23.1.10 Addition of Toggle Button Control

Toggle Button Control is used to change the value of a Register, System or KAREL Variable (except XYZWPR type) or I/O to the specified value following the ON (Pushed)/OFF (Popped) status of the button. Use procedure 23-9 to add Toggle Button Control.

Procedure 23-9 Addition of Toggle Button Control

Step

1 Select Toggle Button in *i*Pendant Control selection screen (Fig. 23.2.2).
2 Press of F4, NEXT leads you to data type setup screen. Please refer to section 23.2.4.
3 Press F4, NEXT. You will see a screen similar to the following.



Create Operation Panel Wizard

Set the type and values of the control:

Type: Logical

False Value:

True Value:

Other Phase:

Caption:

True Caption: (optional)

Toggle Button Control

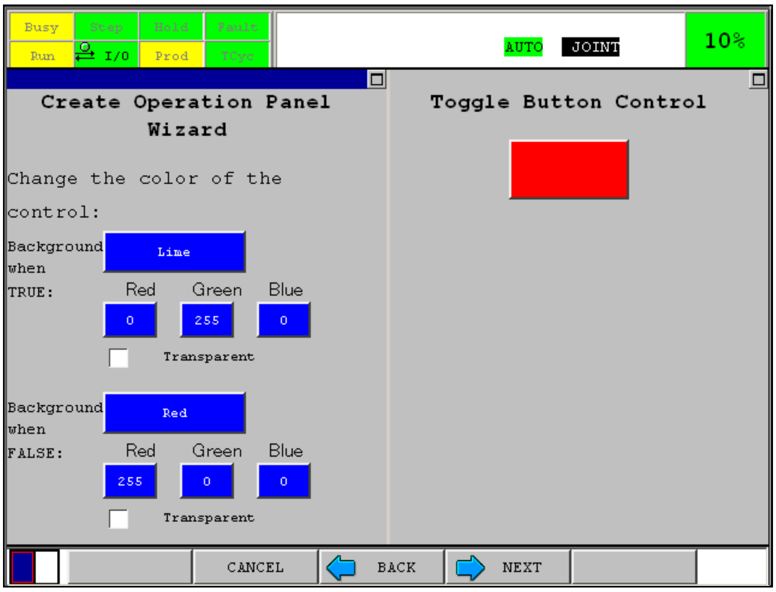
Fig. 23.1.10 (a) Toggle button setup screen

Table 23.1.10 (a) Items in toggle button setup screen

Items	Description
Type	This specifies data type to process. Logical: Data is transacted as boolean type. False value and True Value are not used. Numerical: Data is transacted as numerical type. False Value and True Value are used.
False Value	This is active when Type is Numerical. This specifies the value written when the button status is changed to ON (TRUE).
True Value	This is active when Type is Numerical. This specifies the value written when the button status is changed to OFF (FALSE).
Other Phase	This is active when Type is Numerical. This specifies the status (TRUE/FALSE) in case that the value is equal to neither TrueValue nor FalseValue.
Caption	This specifies the fixed String.
True Caption (optional)	This specifies the fixed String when the value is TRUE. If not specified, Caption is used.

4 Press of F4, NEXT leads you to Toggle Button Image setup screen. Please refer to section 23.3.5.

5 Press F4, NEXT. The following screen will be displayed.



Create Operation Panel Wizard

Change the color of the control:

Background when TRUE: Lime

Red 0 Green 255 Blue 0

□ Transparent

Background when FALSE: Red

Red 255 Green 0 Blue 0

□ Transparent

Toggle Button Control

Fig. 23.1.10 (b) True/False color setup screen

Table 23.1.10 (b) Items in True/False color setup screen

Item	Description
Background when TRUE	This specifies the color displayed when the read value is equal to TrueValue or not equal to FalseValue. How to setup is same as color setup screen. Please refer to section 23.3.6.
Red, Green, Blue	Role is same as color setup screen. Please refer to section 23.3.6.
Transparent	Role is same as color setup screen. Please refer to section 23.3.6.
Background when FALSE	This specifies the color displayed when the read value is equal to FalseValue or not equal to TrueValue. How to setup is same as color setup screen. Please refer to section 23.3.6.

6 Press of F4, NEXT leads you to color setup screen. Please refer to Section 23.2.6.
7 Press of F4, NEXT leads you to size setup screen. Please refer to Section 23.2.3. Caption is used as fixed string. It is not used as format.
8 Press of F4, NEXT leads you to style setup screen. Please refer to Section 23.2.6.
9 Press of F4, NEXT lead you to control insertion screen. Please refer to Section 23.2.3.
10 Decide position to insert and press F4, NEXT. This is end of addition of Toggle Button Control.

Setup of Fast Switch includes setup of data type (step2), image (step 4) and control insertion (step 9) only. Default value is used for items in the other steps. You can add Toggle Button Control by Fast Switch and then modify it.

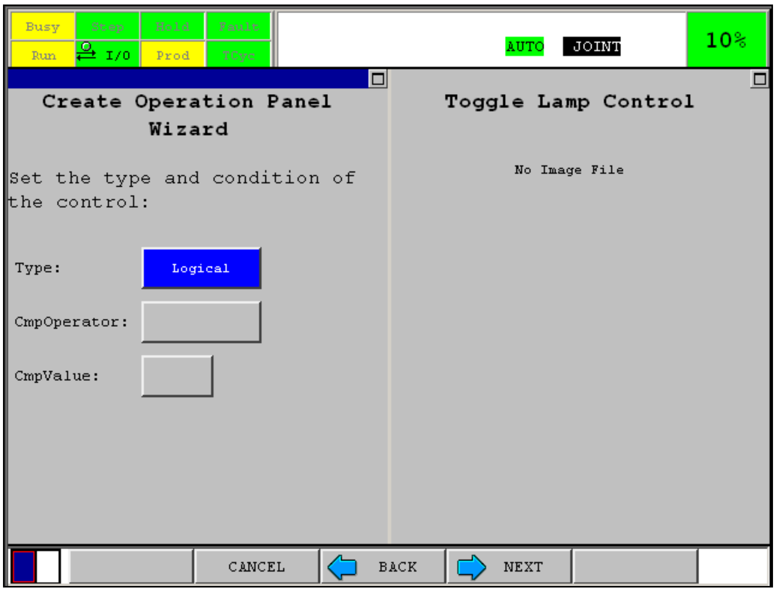
23.1.11 Addition of Toggle Lamp Control

Toggle Lamp Control is used to change the color of the control if the value of a Register, System or KAREL Variable (except XYZWPR type) or I/O fulfills the specified condition with the specified value or not. Use procedure 23-10 to add Toggle Lamp Control.

Procedure 23-10 Addition of Toggle Lampe Control

Step

1 Select Toggle Lamp in *i*Pendant Control selection screen (Fig. 23.2.2).
2 Press of F4, NEXT leads you to data type setup screen. Please refer to Section 23.1.4.
3 Press F4, NEXT. The following screen will be displayed.



Create Operation Panel Wizard

Set the type and condition of the control:

Type: Logical

CmpOperator:

CmpValue:

Toggle Lamp Control

No Image File

Fig. 23.1.11 Toggle lamp setup screen

Table 23.1.11 (a) Items in toggle lamp setup screen

Item	Description
Type	This specifies data type to process. Logical: Data is transacted as bool type. Numerical: Data is transacted as numerical type. The setting of CmpOperator and the value of CmpValue is applied.
CmpOperator	This is active when Type is Numerical. This selects the condition expression evaluated as TRUE. EQ: Specify the equal case (=). NE: Specify the not equal case (<>). LT: Specify the less than case (<). LE: Specify the less than or equal case (<=). GT: Specify the greater than case (>). GT: Specify the greater than or equal case (>=).
CmpValue	This specifies the standard value for comparison. Result of comparison between read value and this value decides status of lamp, TRUE or FALSE. It is reflected to display of lamp.

4 Press of F4, NEXT leads you to image setup screen. Please refer to Section 23.1.4.
5 Press of F4, NEXT leads you to Fig. 23.2.10 (b): True/False color setup screen (Fig. 23.1.10 (b)). Please refer to Section 23.1.10.
6 Press of F4, NEXT leads you to color setup screen. Please refer to Section 23.1.6.
7 Press of F4, NEXT leads you to size setup screen. Please refer to Section 23.1.3. Caption is used as fixed string. It is not used as format.
8 Press of F4, NEXT leads you to style setup screen. Please refer to Section 23.1.6.
9 Press of F4, NEXT lead you to control insertion screen. Please refer to Section 23.1.3.
10 Decide position to insert and press F4, NEXT. This is end of addition of Toggle Lamp Control.

Setup of Fast Lamp includes setup of data type (step2), image (step4) and control insertion (step9) only. Default value is used for items in the other steps. You can add Toggle Lamp Control by Fast Lamp and then modify it.

Background color used by Toggle Lamp depends on its View type.

Table 23.1.11 (b) Background color of Toggle Button

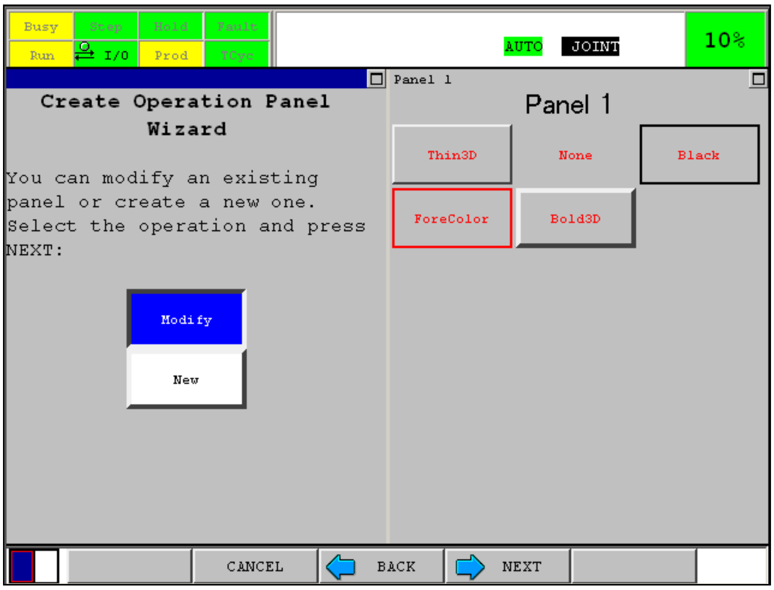
View type	Status	Background color
Panel	True	Background when TRUE
	False	Background when FALSE
Circle, Fixed Circle	True	Circle part: Background when TRUE (If transparent is checked, black) The other part: Background
	False	Circle part: Background when TRUE (If transparent is checked, black) The other part: Background
Image	Both true and false	Image uses background color if it uses transparent color. 1 Background when TRUE (False) if transparent is not checked 2 Background if transparent is not checked 3 Otherwise background color of web page

23.1.12 Modification of Panel

Use procedure 23-11 to modify created panel.

Procedure 23-11 Modification of panel

1 Start panel wizard by procedure 23-1 and select panel.
2 If selected panel already exists, the following screen will be displayed.

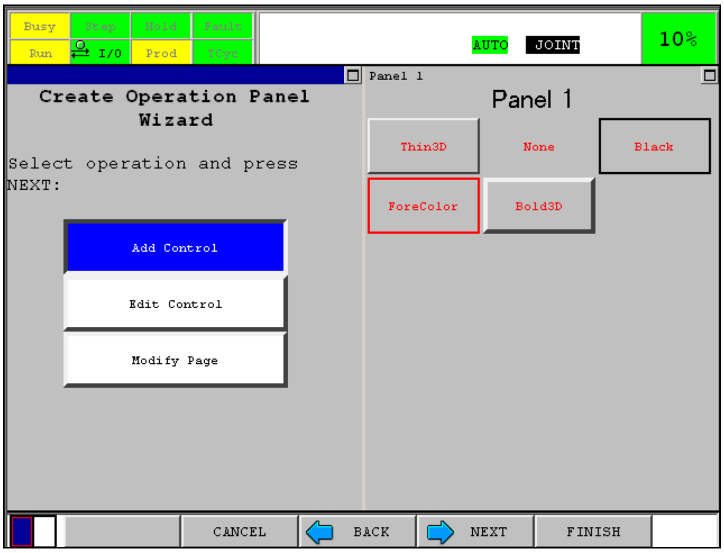


Create Operation Panel Wizard

You can modify an existing panel or create a new one. Select the operation and press NEXT:

Fig. 23.1.12 (a) Selection of modification or creation

3 Select modify and press F4, NEXT. You will see a screen similar to the following.



Create Operation Panel Wizard

Select operation and press NEXT:

Fig. 23.1.12 (b) Selection of operation screen

Table 23.1.12 Items in selection of operation screen

Item	Description
Add Control	You can add control to panel. Please refer to section for control to add.
Edit Control	You can modify control already added to panel. Please refer to section 23.3.13.
Modify Page	You can modify other property of panel. For example, title of panel, background color. Please refer to section 23.3.16.

4 Select operation and press F4, NEXT.

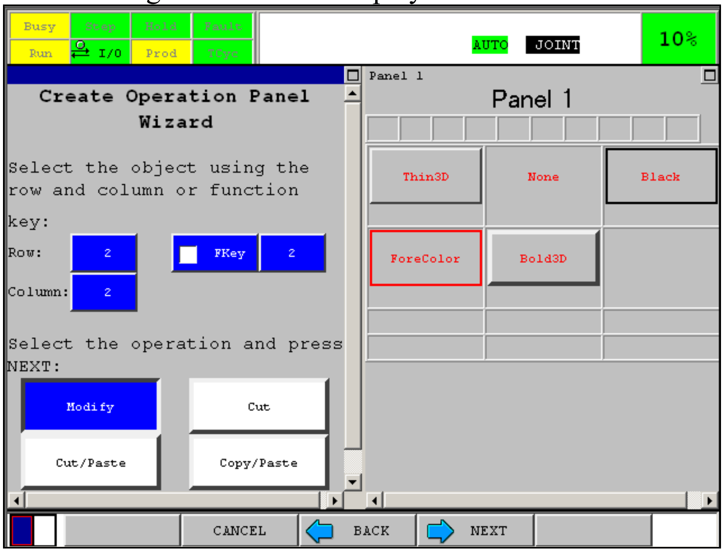
For detail of each operation, please refer to section written in table above.

23.1.13 Modification of Control

Use procedure 23-12 to modify control that is already added to panel.

Procedure 23-12 Modification of control

1 Display selection of operation screen by procedure 23-11.
2 Select Edit Control.
3 Press F4, NEXT. The following screen will be displayed.



Create Operation Panel Wizard

Select the object using the row and column or function key:

Row: 2

Column: 2

□ FKey 2

Select the operation and press NEXT:

Fig. 23.1.13 Edit control screen

Table 23.1.13 Items in edit control screen

Item	Description
Row, Column	These specify control to modify, cut, and copy.
Fkey	If this is checked, control assigned to function key is selected.
The number next to Fkey	This number specifies function key when Fkey is checked.
Modify	Select this to modify control already added to panel.
Cut	Select this to delete control.
Cut/Paste	Select this to cut selected control and paste it elsewhere.
Copy/Paste	Select this to copy selected control and paste it elsewhere.

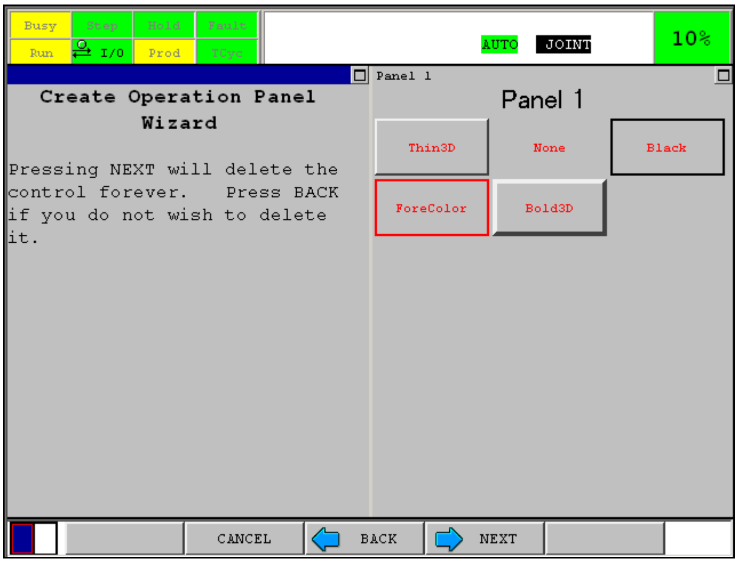
4 Select Modify and press F4, NEXT.
5 Setup screen appropriate for the selected control is displayed. If you added the control by Fast label, Fast switch or Fast lamp, setup screen for corresponding control is displayed. For example, setup screen for Label control is displayed for control added by Fast label.
6 Setup control just same as addition of control. Control insertion screen at the last step is not displayed.

23.1.14 Delete of Control

Use procedure 23-13 to delete control.

Procedure 23-13 Delete of control

1 Display Edit control screen by procedure 23-12.
2 Select control to delete.
3 Select Cut and press F4, NEXT.
4 The following screen will be displayed.



Create Operation Panel Wizard

Pressing NEXT will delete the control forever. Press BACK if you do not wish to delete it.

Fig. 23.1.14 Confirmation of delete of control

Press F4 to delete. Otherwise, press F3.

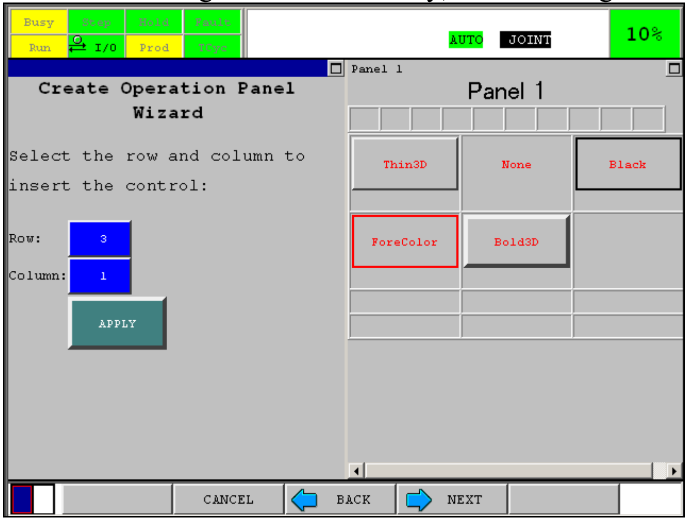
> **NOTE**
> F2 CANCEL cancels panel wizard. Press of F2 in this screen causes delete of control.

23.1.15 Cut/Copy Paste of Control

You can move added control by cut and paste. You can copy added control by copy and paste. Use procedure 23-14.

Procedure 23-14 Cut/Copy paste of control

1 Display Edit control screen by procedure 23-12.
2 Select control.
3 If you want to move the control, select Cut/Paste. If you want to copy it, select Copy/Paste. Press F4, NEXT.
4 Next screen depends on control you selected.
4-1 If selected control is NOT assigned to function key, the following screen will be displayed.



Create Operation Panel Wizard

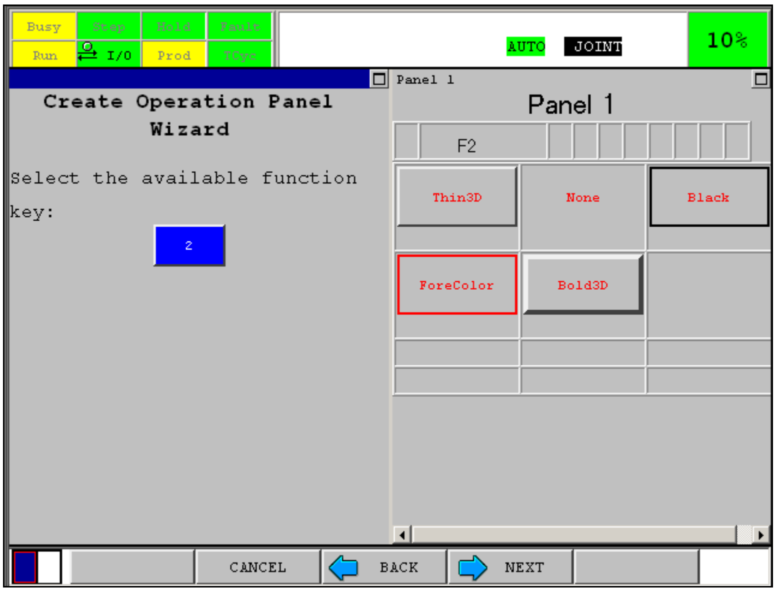
Select the row and column to insert the control:

Row: 3

Column: 1

Fig. 23.1.15 (a) Control insertion screen

4-2 If selected control is assigned to function key, the following screen will be displayed. This screen is to select function key to paste control.



Create Operation Panel Wizard

Select the available function key:

2

Fig. 23.1.15 (b) Control insertion screen (function key)

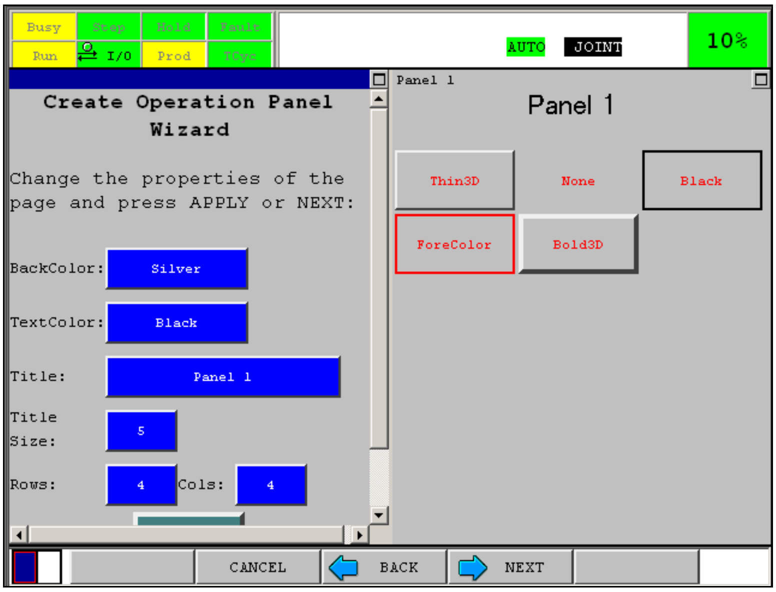
5 Decide position to insert control and Press F4, NEXT.

23.1.16 Modification of Page

Use procedure 23-15 to modify property of page (panel). Background color and title for example.

Procedure 23-15 Modification of page

1 Display selection of operation screen by procedure 23-11.
2 Select Modify page.
3 Press F4, NEXT. The following screen will be displayed.



Create Operation Panel Wizard

Change the properties of the page and press APPLY or NEXT:

BackColor: Silver

TextColor: Black

Title: Panel 1

Title Size: 5

Rows: 4 Cols: 4

Fig. 23.1.16 Page edit screen

Table 23.1.16 Item in page edit screen

Item	Description
BackColor	Background color of page
Text Color	Color of text. Title of color uses this.
Title	This is title displayed at top of panel. If panel1, it defaulted to Panel 1.
Title Size	Size of title. Range is from 1 to 5.
Rows	Control is placed in invisible table. This item is the number of rows of the table. You cannot decrease it.
Cols	The number of columns of invisible table. You cannot decrease it.
APPLY	This is not item to setup. Press of the button applies property edited and you can see rough image in right pane.

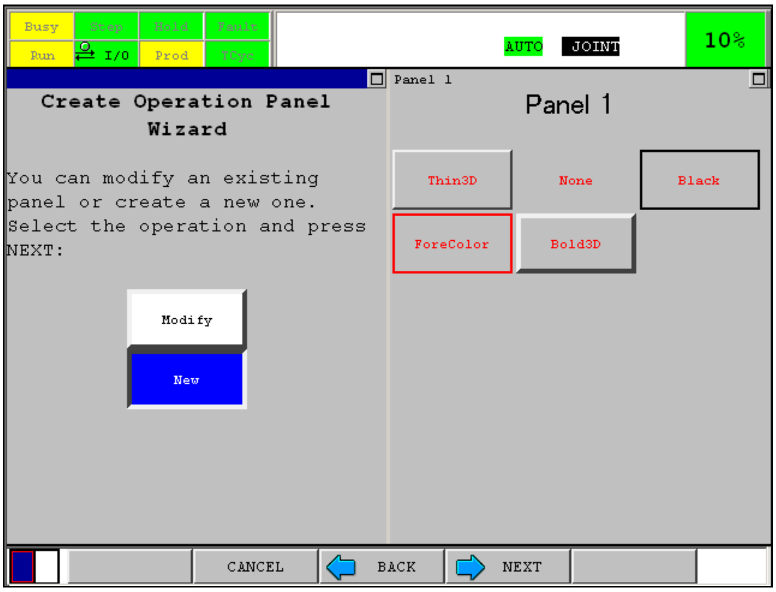
4 Setup as required and Press F4, NEXT. This is the end of modification of page.

23.1.17 Re-creation of Panel

Use procedure 23-16 to re-create panel with disposing existing panel.

Procedure 23-16 Re-creation of panel

1 Display panel selection screen by procedure 23-1.
2 Select panel to re-create and press F4, NEXT.
3 Screen to select modification or creation is displayed.
4 Select New.

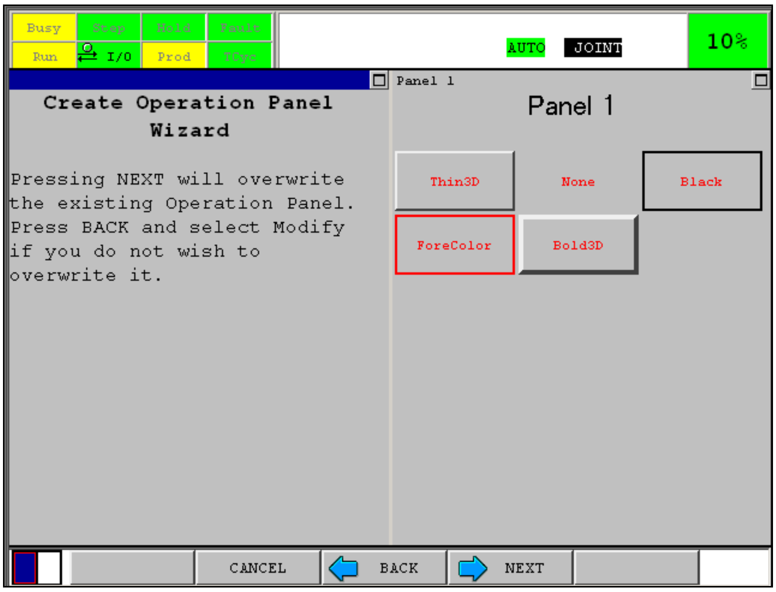


Create Operation Panel Wizard

You can modify an existing panel or create a new one. Select the operation and press NEXT:

Fig. 23.1.17 (a) Selection of modification or creation

5 Press F4, NEXT. The following screen will be displayed.



Create Operation Panel Wizard

Pressing NEXT will overwrite the existing Operation Panel. Press BACK and select Modify if you do not wish to overwrite it.

Fig. 23.1.17 (b) Confirmation of re-creation of panel

6 Press of F4 delete existing panel. Re-creation starts from setup of the number of rows and columns.

23.2 RUN KAREL PROGRAM BY PANEL

You can create button to run KAREL program. Required steps are following.

(A) Creation of KAREL program that can be run by panel

(B) Addition and setup of Button Change Control

NOTE

You cannot run arbitrary KAREL program. There is caution of creation of KAREL program for the program to be run by button change control.

23.2.1 Caution for Creation of KAREL Program

(A) Create static (program level) INTEGER variable, return_code. Set it to 204.

(B) Program must complete in a short time.

(C) If it takes long time to complete, run program by RUN_TASK built-in by another program that was run by button change control.

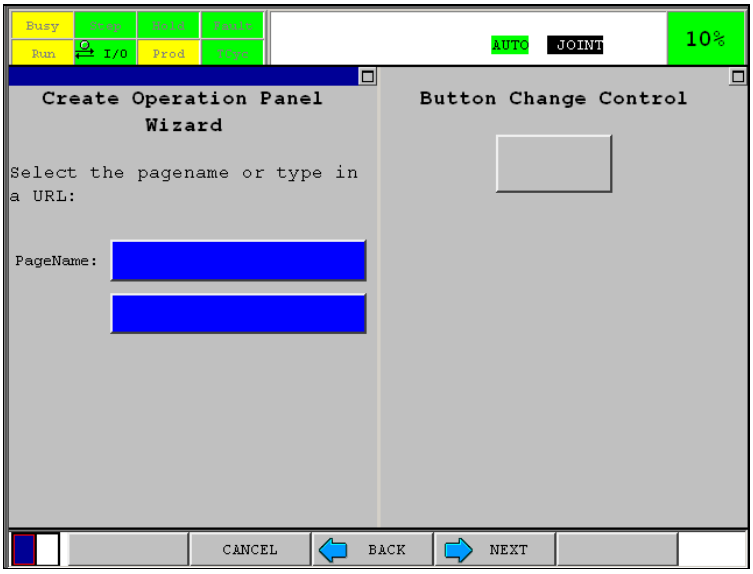
(D) Specify %NOLOCKGROUP.

23.2.2 Creation of Run Button

Use procedure 23-17 to create button to run program.

Procedure 23-17 Creation of run button

1 Start addition of Button Change Control and display URL setup screen.
Please refer to Subsection 23.1.6 for procedure to add Button Change control.



Create Operation Panel Wizard

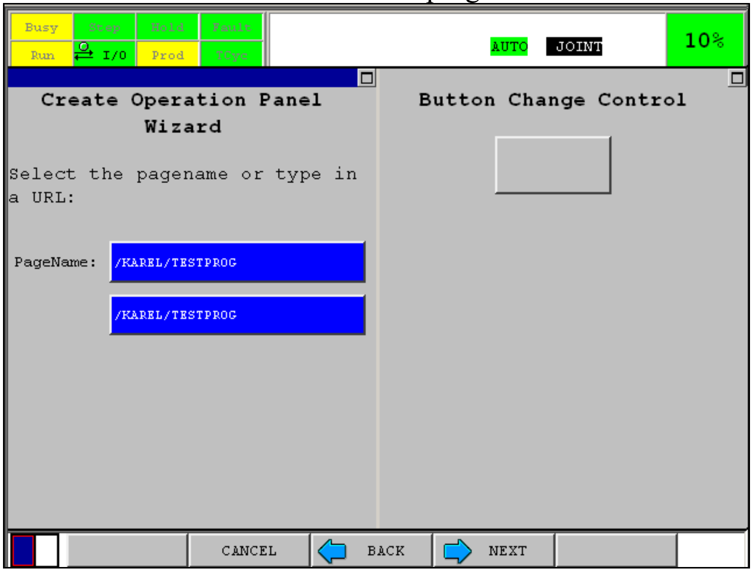
Select the pagename or type in a URL:

PageName:

Button Change Control

Fig. 23.2.2 (a) URL setup screen

2 Press lower side button and input URL in format of “/KAREL/program name”.
For example, /KAREL/TESTPROG to run KAREL program TESTPROG.



Create Operation Panel Wizard

Select the pagename or type in a URL:

PageName: /KAREL/TESTPROG

/KAREL/TESTPROG

Button Change Control

Fig. 23.2.2 (b) Setup to run TESTPROG

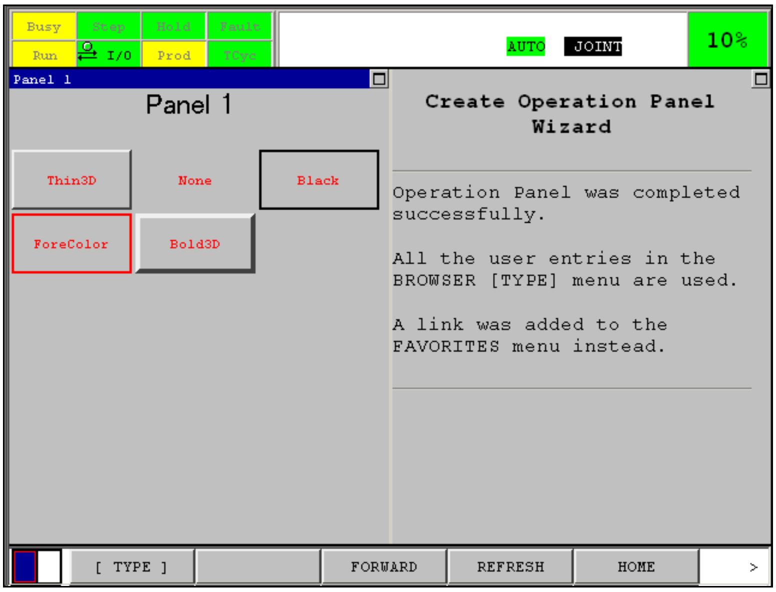
23.3 USAGE

23.3.1 Display of Panel

MENU→BROWSER→[TYPE] displays panels created.

> **NOTE**
> Created panels are used in all languages. If you display the panel in more than one language, create the panel in English. Japanese panel is not displayed correctly in English.

If there are too many entries in [TYPE] menu of BROWSER to add panel to the menu, you will see a screen similar to following at finishing addition of panel.



Create Operation Panel Wizard

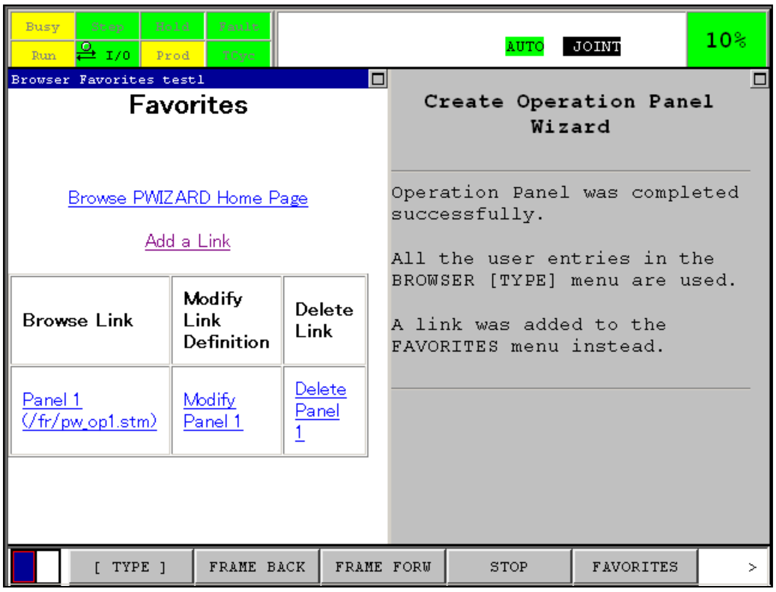
Operation Panel was completed successfully.

All the user entries in the BROWSER [TYPE] menu are used.

A link was added to the FAVORITES menu instead.

Fig. 23.3.1 (a) Notification of addition to favorite

In this case, panel is added to link in FAVORITE. Display panel by selecting link.



Favorites

Browse PWIZARD Home Page

Add a Link

Browse Link	Modify Link Definition	Delete Link
Panel 1 (/fr/pw_op1.stm)	Modify Panel 1	Delete Panel 1

Create Operation Panel Wizard

Operation Panel was completed successfully.

All the user entries in the BROWSER [TYPE] menu are used.

A link was added to the FAVORITES menu instead.

Fig. 23.3.1 (b) Panel1 added to favorite

23.3.2 Backup/Restore

Created panels can be saved and restored in file screen. It is categorized as “Application”.
Panel1…Panel 4 corresponds to PW_OP1.STM…PW_OP4.STM, respectively.

NOTE

PANEL1.STM,PANEL1.DT,……,PANEL4.STM, PANEL4.DT have nothing to do with this function.

- If you use image file you prepared, use it in GIF or JPG format. Copy them to FR: and refer to it. FR: *.GIF and FR: *.JPG are saved and restored in file screen. They are also in category of “Application”.

- Registration data to menu of MENU→BROWSER→[TYPE] is saved in SYSVARS.SV. If you want to backup/restore the registration, SYSVARS.SV should be backed up and restored.

- If you do not want to restore SYSVARS.SV, you have to register panel to [TYPE] menu manually. Use procedure below.
  - Restore backup of panel. (PW_*.STM and required image files)
  - Display Selection of operation screen (Fig. 23.1.12 (b)) by procedure 23-11.
  - Press F5, FINISH. This means there is actually no change.
  - These steps add panel to [TYPE] menu of BROWSER.

24 ENHANCED MIRROR IMAGE

The enhanced mirror image option (A05B-2600-R698) allows you to translate an entire teach pendant program or portion of a teach pendant program to mirror-image the original programmed points. This option can be used to teach symmetrical parts easily. The enhanced mirror image option provides additional functionality so that existing frames such as user frames can be used directly with any specified XY, YZ, or XZ plane. Also, additional orientation shifting is provided for different application requirements.

> ⚠ **NOTE**
> For Paint Tool and paint robots, this is almost always accomplished using left-hand and right-hand robots. Use of this option is not recommended.

Mirror imaging of a program can be accomplished either as:

- mirror image using existing frames such as World frame or user frames. No teaching required.
- mirror image using flexible mirror planes such as X-Y, Y-Z, or X-Z from existing frames for better selection
- parallel mirror image with mirrored orientation
- parallel and rotational mirror image with mirrored orientation
- mirror image orientation shifting can be selected based on application requirements.

Refer to Table 24 for descriptions of the Mirror Image Shift screen items.

Table 24 ENHANCED MIRROR IMAGE screen items

ITEM	DESCRIPTION
Original Program	This item is the name of the program that you want to shift using mirror image. To select a program, press F4, [CHOICE], select the name of program, and press ENTER again.
Range	This item indicates whether the whole program (F5, WHOLE) or a portion (F4, PART) will be shifted using mirror image.
Start line (used only for “PART” range)	This item is the first line in the program that will be shifted using mirror image.
End line (used only for “PART” range)	This item is the last line in the program that will be shifted using mirror image.
New Program	This item is the name of the program that results when you offset or shift the Original Program. If you want the resulting offset or shifted program to insert into the Original Program, make the New Program name the same as the Original Program name.
Insert line	This item indicates the location in an existing program where new shifted position information will be inserted.
Position data (used only for “TEACH” method)	This item is the current x, y, and z position information.
Image Shift Method	● F5, TEACH is a method that requires the user to enter position data to define the mirror plane. ● F4, FRAME is a method that allows the user to choose existing frames.
Orientation Shift	● F5, MIRROR is a mirror image shifting method that mirrors the orientation as well as the positions. ● F4, FIXTURE is a mirror image shifting method that only mirrors the approach vector of the orientation but leaves the other two vectors intact with flexible tool orientation control.
Mirror Frame (used only for “FRAME” method)	World, User Frame, or Jog Frame can be selected as the mirror frame.
Frame No. (used only for “FRAME” method)	When User frame or Jog frame is selected, this indicates which frame number will be used.

ITEM	DESCRIPTION
Mirror Plane (used only for “FRAME” method)	The XY, YZ, or XZ plane of the selected mirror frame can be used as mirror plane.
Rotation (used only for “TEACH” method)	This item, if set to ON, indicates that you want to rotate three position about an axis. If set to OFF, you can shift a single source position to a single destination position.
Source position (used only for “TEACH” method)	This item is the original position that you will be shifting using mirror image. If Rotation is set to ON, there will be three source positions.
Destination position (used only for “TEACH” method)	This item is the destination position to which you will be shifting using mirror image. If Rotation is set to ON, there will be three destination positions.

24.1 PARALLEL MIRROR IMAGE

A parallel mirror image mirrors the program about a mirror plane without an offset or a rotation. See Fig. 24.1 (a) and Fig. 24.1 (b).

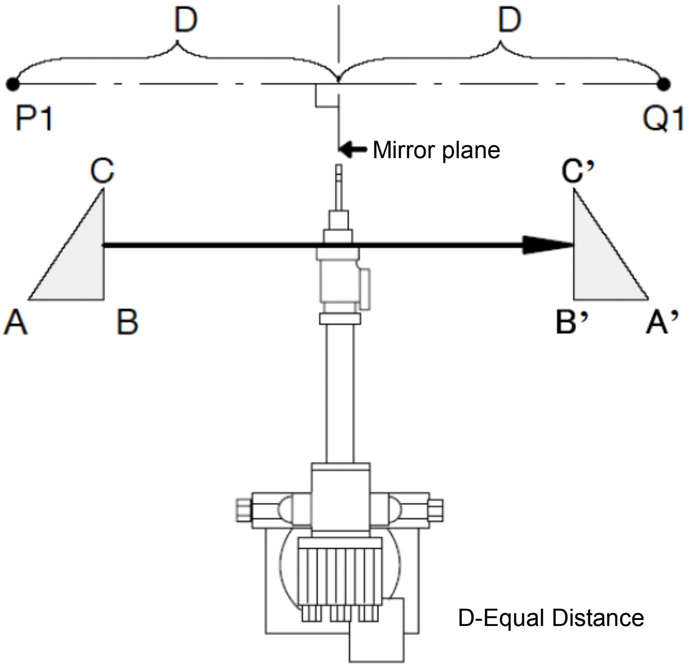


Fig. 24.1 (a) Parallel mirror image with mirror plane in center of robot

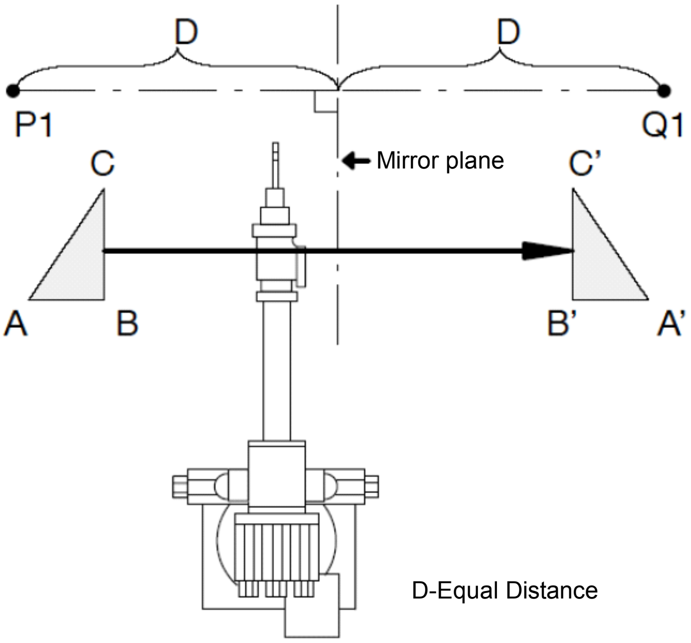


Fig. 24.1 (b) Parallel mirror image with mirror plane offset from center of robot

⚠ **CAUTION**
To be sure the parallel mirror image works correctly, you must have an exact TCP. If you do not, the resulting mirror image program will contain an offset value. See Fig. 24.1 (c).

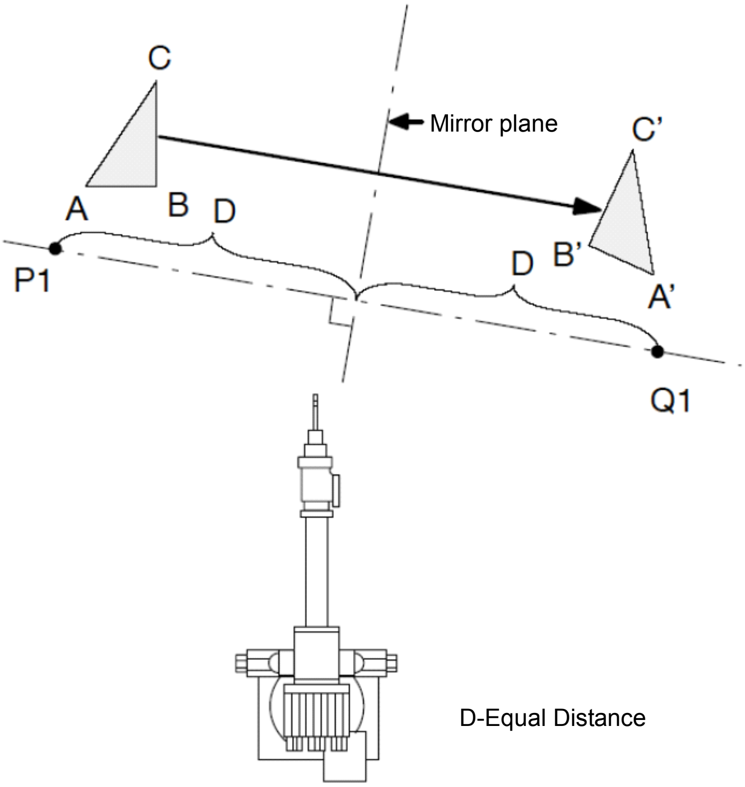


Fig. 24.1 (c) Parallel mirror image with offset

Mirror Image Example

Fig. 24.1 (d) displays a mirror image of A, B, and C to A', B', and C' when P1 and Q1 are taught positions. Fig. 24.1 (d) also displays a mirror image with an offset when P1 and Q1 are taught but Q1 is taught at an offset of 200mm. In this case, the result is A'', B'', and C''.

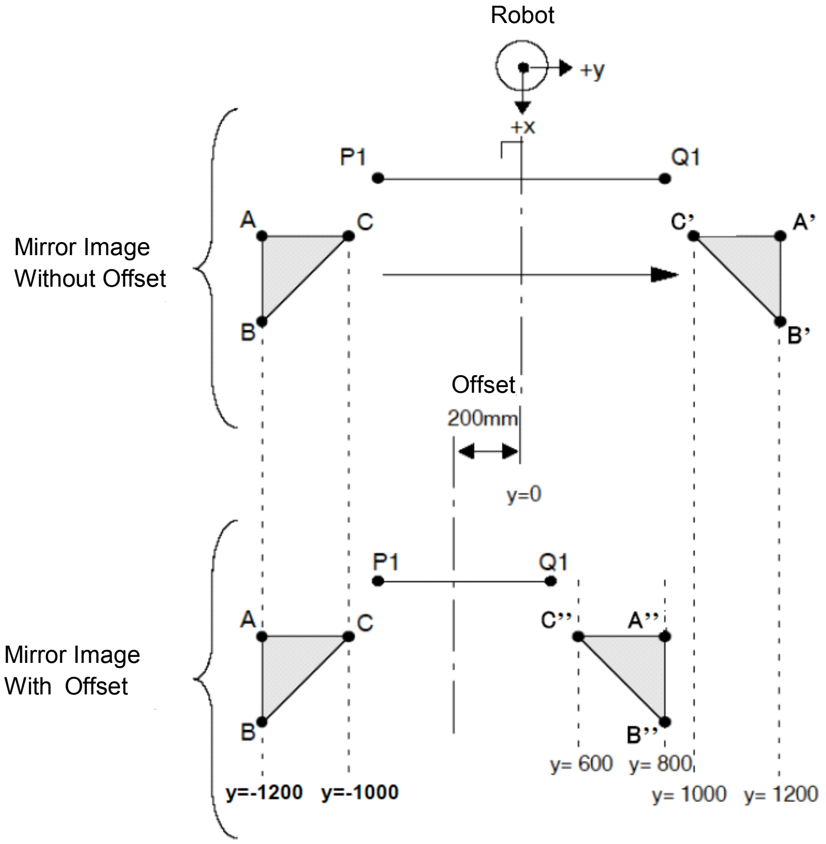


Fig. 24.1 (d) Positional mirror image

24.2 ROTATIONAL MIRROR IMAGE

A parallel mirror image mirrors the program about a mirror plane first, and then the mirrored program is rotated about a center of rotation. Orientation of the part to be imaged is rotated about one or more of its axes relative to the mirror plane. See Fig. 24.2 (a).

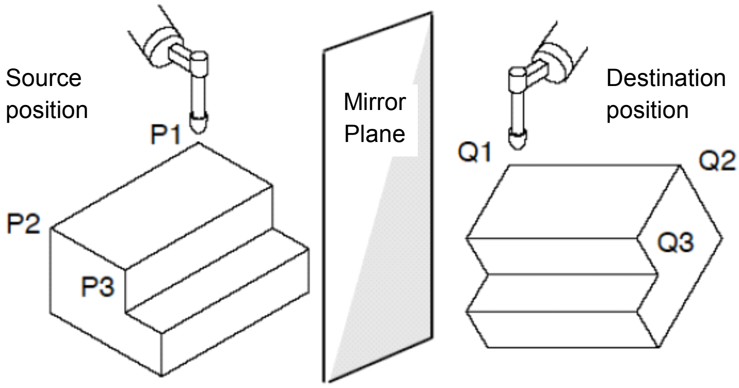


Fig. 24.2 (a) Rational mirror image

In Fig. 24.2 (b) the positions, P1, P2 and P3 are mirrored about the mirror plane as Q1, Q2 and Q3. These positions are then rotated 45° about Q1 and stored as Q1', Q2' and Q3'.

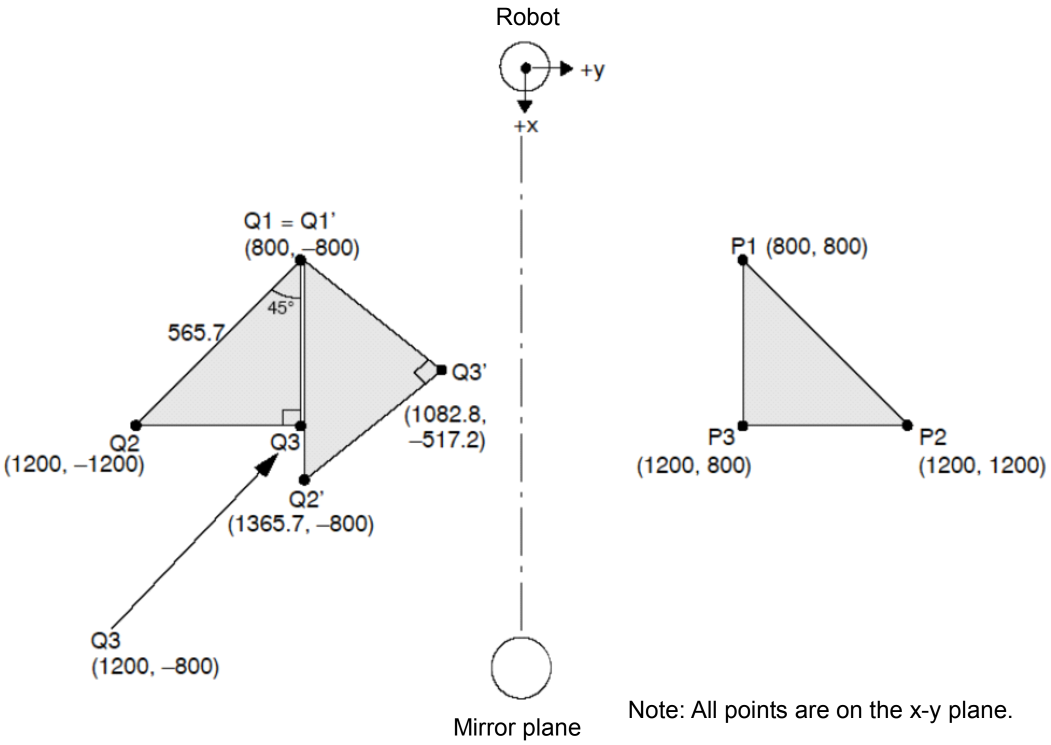


Fig. 24.2 (b) Rational mirror image

24.3 MIRROR IMAGE USING EXISTING FRAMES AND MIRROR PLANES, WITH CONTROLLED ORIENTATION

The standard mirror image utility mirrors orientation as well as positions, as shown in Fig. 24.3 (a).

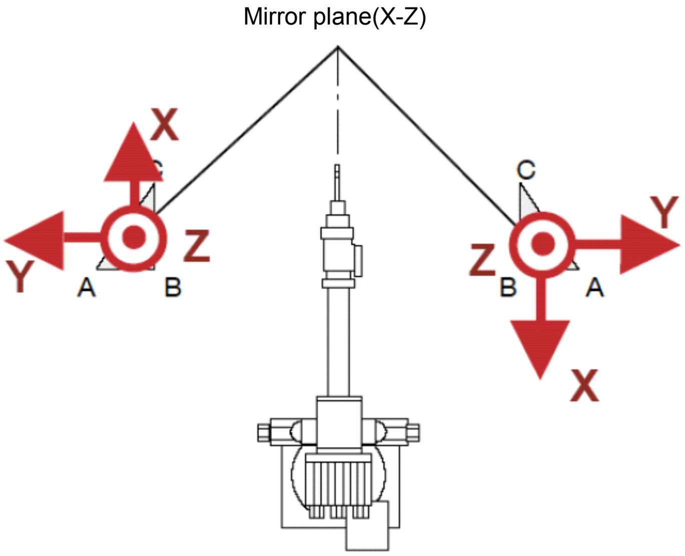


Fig. 24.3 (a) Mirror image shift with orientation mirrored (mirror method)

The enhanced mirror image option also provides a simple way to use existing frames, such as World, User, or Jog Frame, with defined mirror planes, such as X-Y, Y-Z, or X-Z. Also, to control the orientation as well as the positions in the mirrored program, the Orientation Mirror method can be selected as "Mirror" as shown in Fig. 24.3 (a), or "Fixture", as shown in Fig. 24.3 (b).

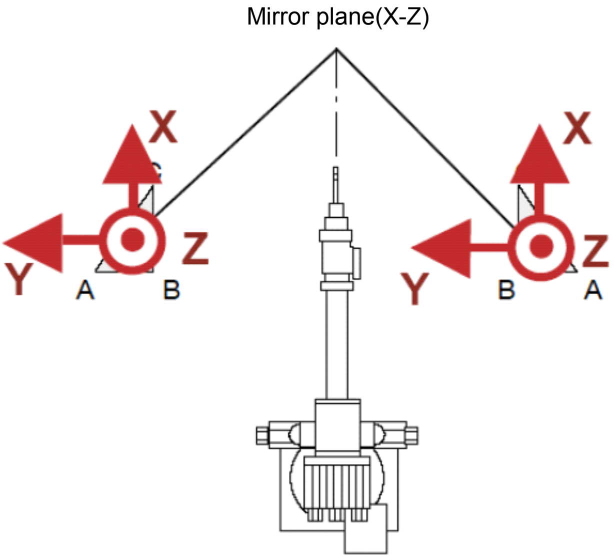


Fig. 24.3 (b) Enhanced mirror image with orientation controlled (fixture method)

24.4 MIRROR IMAGE OF EXTENDED AXES

Extended (Ext) axes determine how the mirror image function will translate the program when you are using extended axes. Fig. 24.4 (a) displays a key for use with Fig. 24.4 (b) through Fig. 24.4 (d).

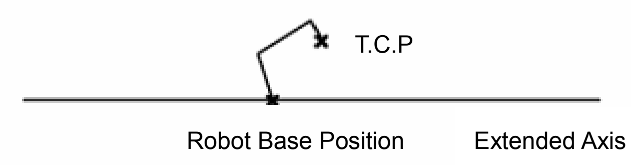


Fig. 24.4 (a) Mirror image key

Possible extended axes configurations are:

- **Robot axes only -** allows you to mirror the axes of the robot without mirroring any non-integrated extended axes such as a positioning table. The shift is calculated using the change in the robot tool center point (TCP). See Fig. 24.4 (b).

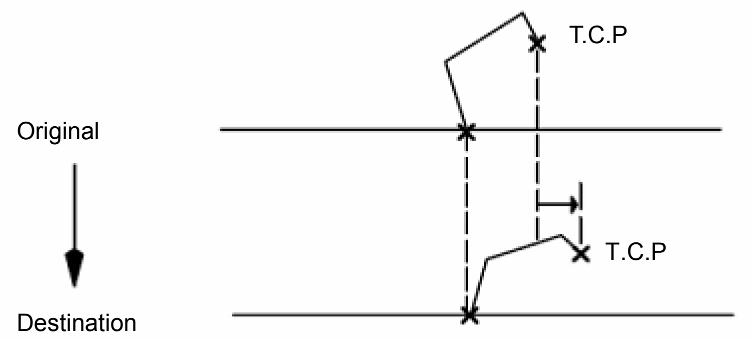


Fig. 24.4 (b) Example of robot axes only mirror image

- **Ext integrated -** allows you to mirror the axes of the robot and any integrated axes. The amount of mirror image for the robot and the extended axes is calculated using the change in the TCP. See Fig. 24.4 (c).

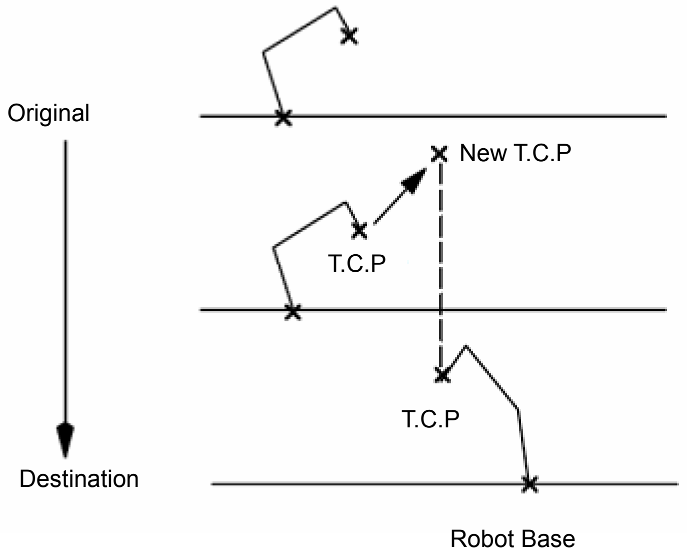


Fig. 24.4 (c) Example of extended axes integrated mirror image

- **With ext axes -** allows you to mirror positions for robot axes and any extended axes in your system. The amount of mirror image for the robot is calculated by using the change in the TCP. The amount of shift for the extended axes is calculated using the center of the difference between an original position (P1) and a new position (Q1) as the point where the mirror image occurs. See Fig. 24.4 (d).

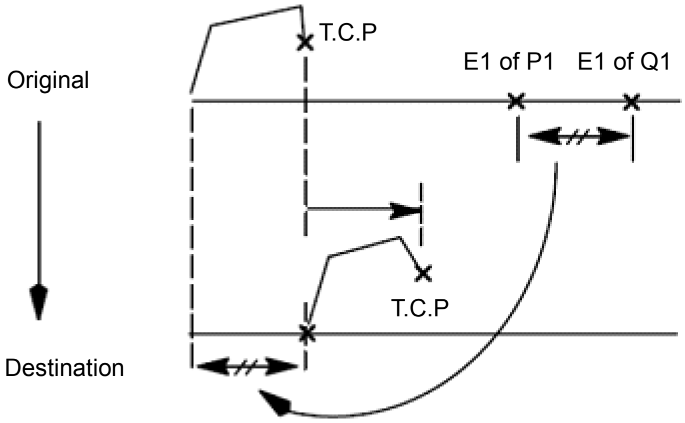


Fig. 24.4 (d) Example of with extended axes mirror image

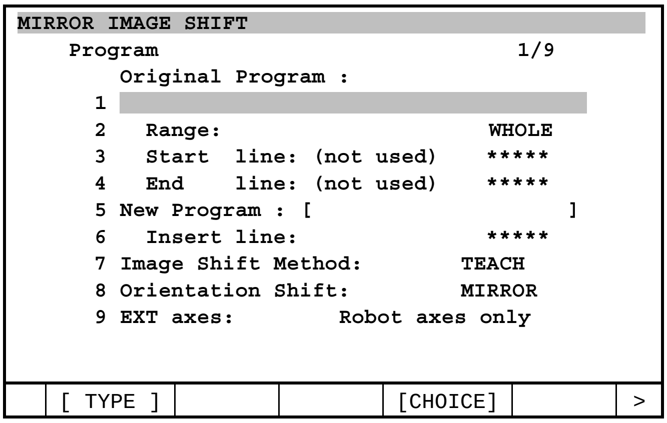
Procedure 24-1 Using Enhanced Mirror Image Shift

Conditions

- The program you want to mirror has been created and contains recorded positions.
- All robot joint axes are at zero degrees.

Steps

1. Press [MENU] key.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select Mirror Image Shift. You will see the MIRROR IMAGE SHIFT screen.



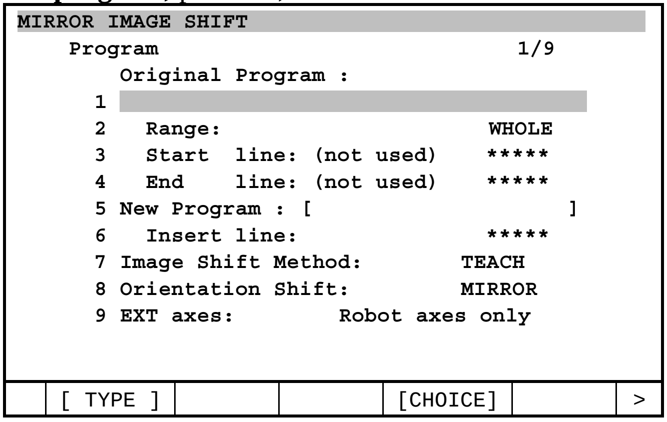
```
MIRROR IMAGE SHIFT
    Program                            1/9
        Original Program :
      1
      2   Range:                        WHOLE
      3   Start  line: (not used)     *****
      4   End    line: (not used)     *****
      5 New Program : [                    ]
      6   Insert line:                *****
      7 Image Shift Method:          TEACH
      8 Orientation Shift:           MIRROR
      9 EXT axes:        Robot axes only

  [ TYPE ]              [CHOICE]         >
```

5. Move the cursor to 1 under Original Program. If the program you want to mirror is not selected, press F4, [CHOICE]. Select the program from the list and press ENTER.

> ⚠ **NOTE**
> The last program selected using the SELECT menu will automatically be named as the original program.

6. Move the cursor to Range. Use the function keys to select whether you want to mirror all or part of the original program.

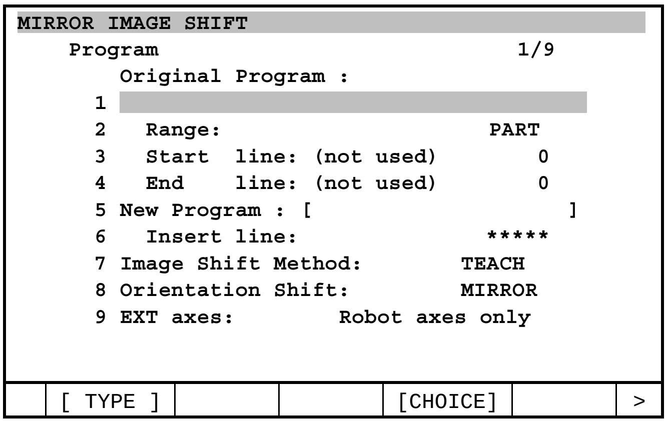
- **To mirror the whole program,** press F5, WHOLE. You will see a screen similar to the following.



```
MIRROR IMAGE SHIFT
    Program                            1/9
        Original Program :
      1
      2   Range:                        WHOLE
      3   Start  line: (not used)     *****
      4   End    line: (not used)     *****
      5 New Program : [                    ]
      6   Insert line:                *****
      7 Image Shift Method:          TEACH
      8 Orientation Shift:           MIRROR
      9 EXT axes:        Robot axes only

  [ TYPE ]              [CHOICE]         >
```

> ⚠ **NOTE**
> EXT axes will be displayed only if you are using mirror image for extended axes.

- **To shift part of the program,** press F4, PART. You will see a screen similar to the following.



```
MIRROR IMAGE SHIFT
    Program                            1/9
        Original Program :
      1
      2   Range:                        PART
      3   Start  line: (not used)         0
      4   End    line: (not used)         0
      5 New Program : [                    ]
      6   Insert line:                *****
      7 Image Shift Method:          TEACH
      8 Orientation Shift:           MIRROR
      9 EXT axes:        Robot axes only

  [ TYPE ]              [CHOICE]         >
```

7. **If you selected to mirror PART of a program,** select the lines of the program that you want to mirror:

- Move the cursor to Start line and type the starting line number. Press ENTER.
- Move the cursor to End line and type the ending line number. Press ENTER.

8. Move the cursor to New Program and press ENTER. Use the appropriate function keys to type the name of the new program and press ENTER. This is the program to which you will be mirroring the positions.

> ⚠ **NOTE**
> You can mirror the positions of an entire program or portion of a program from within a program or from one program to another. See Fig. 24.4 (e) and Fig. 24.4 (f).

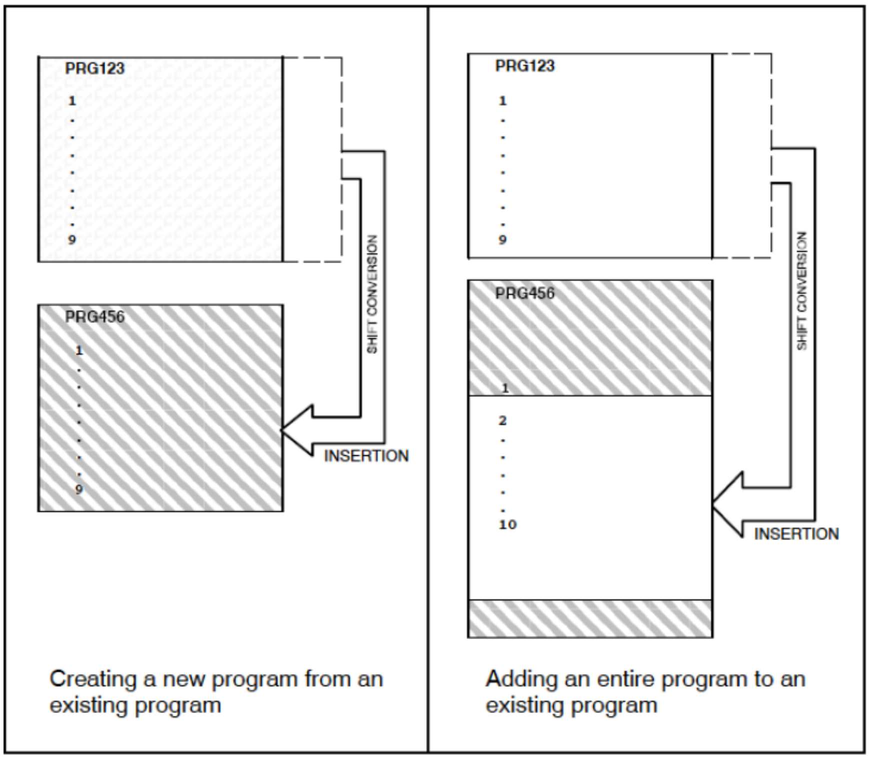

PRG123
1
9
PRG456
1
9
SHIFT CONVERSION
INSERTION
Creating a new program from an existing program
PRG123
1
9
PRG456
1
2
10
SHIFT CONVERSION
INSERTION
Adding an entire program to an existing program

Fig. 24.4 (e) Mirroring an entire program

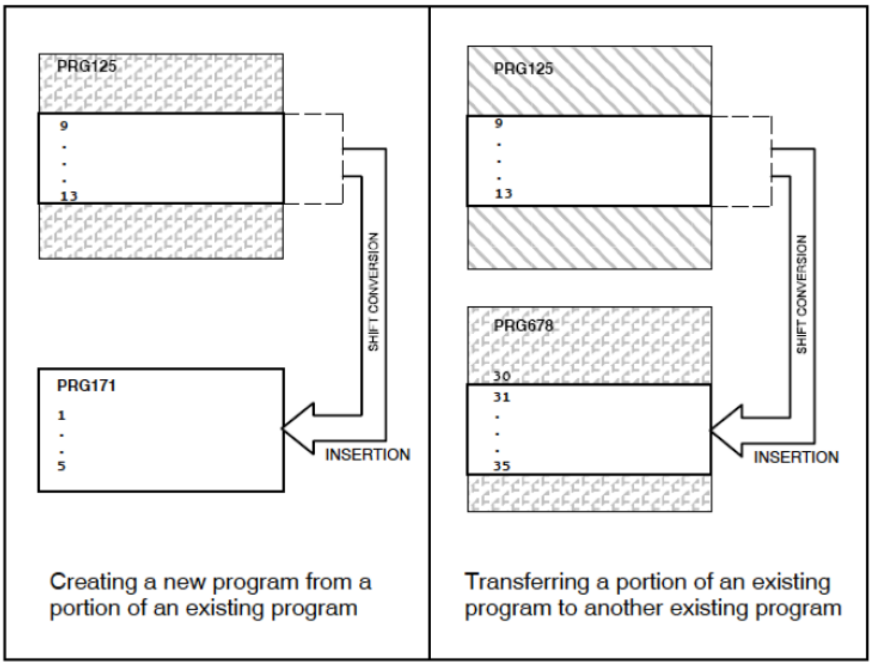
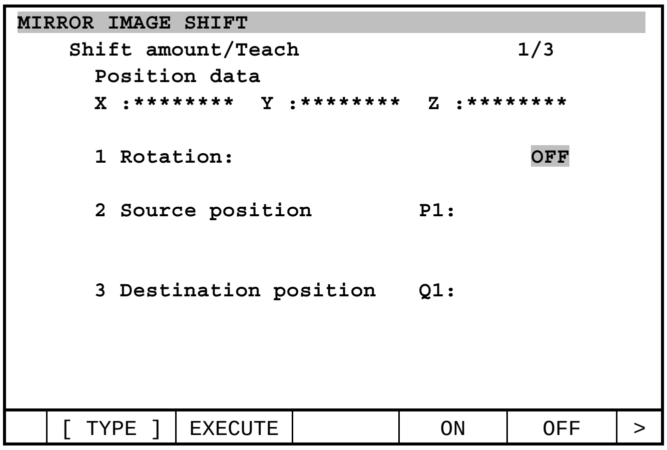

PRG125
9
13
PRG171
1
5
SHIFT CONVERSION
INSERTION
Creating a new program from a portion of an existing program
PRG125
9
13
PRG678
30
31
35
SHIFT CONVERSION
INSERTION
Transferring a portion of an existing program to another existing program

Fig. 24.4 (f) Mirroring a portion of a program

> ⚠ **NOTE**
> The New Program can be the same as the original program, an already existing program, or a program that does not exist.

- **If you are inserting lines into the original program,** type in the name of the program and the corresponding line numbers.
- **If you are inserting lines into an existing program,** move the cursor to Insert line and type the line number at which you want to insert the shifted information.

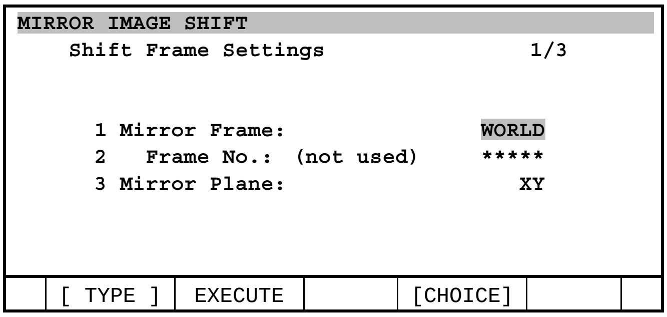
9. Move the cursor to Image Shift Method.
   - **To mirror the image using existing User, Jog, or World frames** and well-defined mirror frames such as Z-Y, Y-Z or X-Z, press F4, FRAME.
   - **To mirror the image by teaching points to define the mirror plane,** press F5, TEACH. This method also mirrors the orientation.
10. Move the cursor to Orientation Shift.
    - **To control the orientation shift** in the mirrored program, press F4, FIXTURE.
    - **To mirror the orientation** of the original program, press F5, MIRROR.
11. Press the down arrow key.
    - **If you selected TEACH** for the Image Shift Method, you will see a Shift amount/Teach screen similar to the following:



```
MIRROR IMAGE SHIFT
    Shift amount/Teach                1/3
      Position data
      X :********  Y :********  Z :********

      1 Rotation:                       OFF

      2 Source position        P1:


      3 Destination position   Q1:
```

If you do not want to rotate the positions, continue to Step 12.
If you want to rotate the positions, continue to Step 13.

- **If you selected FRAME** for the Image Shift Method, you will see a Shift Frame Settings screen similar to the following:

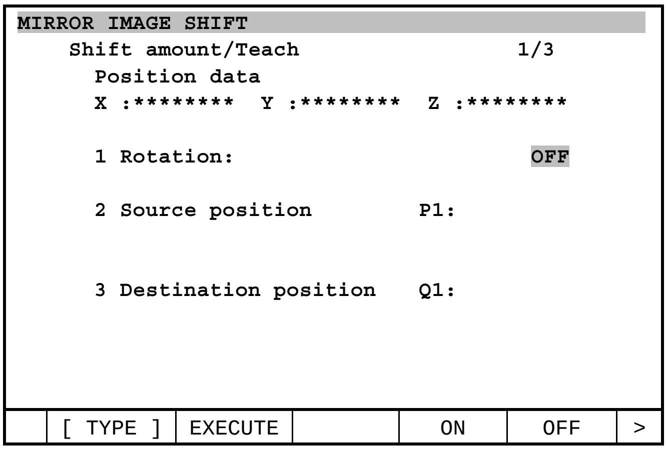


```
MIRROR IMAGE SHIFT
    Shift Frame Settings                1/3


      1 Mirror Frame:                 WORLD
      2   Frame No.:  (not used)      *****
      3 Mirror Plane:                    XY
```

Continue to Step 14.

12. If you do not want to rotate the positions,

a. Press F5, OFF. The following screen will be displayed.

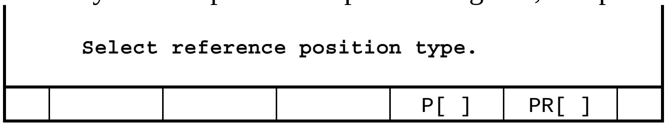


```
MIRROR IMAGE SHIFT
    Shift amount/Teach                1/3
      Position data
      X :********  Y :********  Z :********

      1 Rotation:                       OFF

      2 Source position        P1:


      3 Destination position   Q1:



  [ TYPE ]  EXECUTE            ON       OFF      >
```

b. Move the cursor to source position (see Fig. 24.1 (a)).

c. Move the robot to the source position (P1) and either record or specify the position:

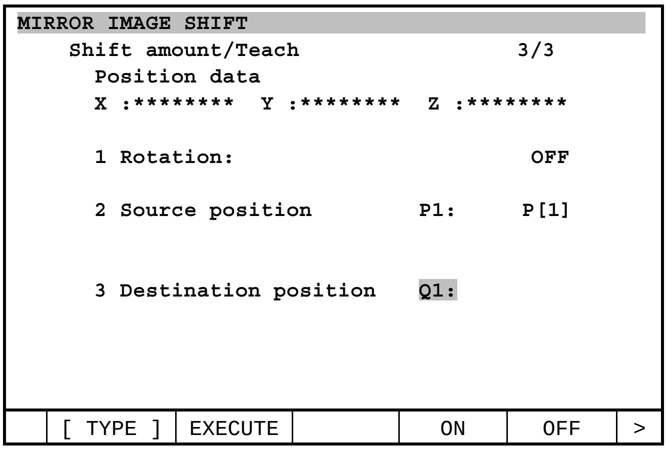
- **To record a position,** jog the robot to the position you want, press and hold in [SHIFT] key and press F5, RECORD.
- **To specify a previously recorded position or position register,** press F4, REFER. Type the number of a previously defined position or position register, and press [ENTER] key.



```
     Select reference position type.

                                  P[ ]     PR[ ]
```

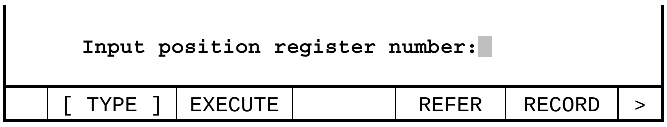
- **To clear a position setting,** press [NEXT] key, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

d. Move the cursor to Destination position. The following screen will be displayed.



```
MIRROR IMAGE SHIFT
    Shift amount/Teach                3/3
      Position data
      X :********  Y :********  Z :********

      1 Rotation:                       OFF

      2 Source position        P1:     P[1]


      3 Destination position   Q1:



  [ TYPE ]  EXECUTE            ON       OFF      >
```

- **To record a position,** jog the robot to the destination position (Q1). Press and hold in [SHIFT] key and press F5, RECORD.
- **To specify a previously recorded position register,** press F4, REFER. Type the number of the position register and press [ENTER] key.



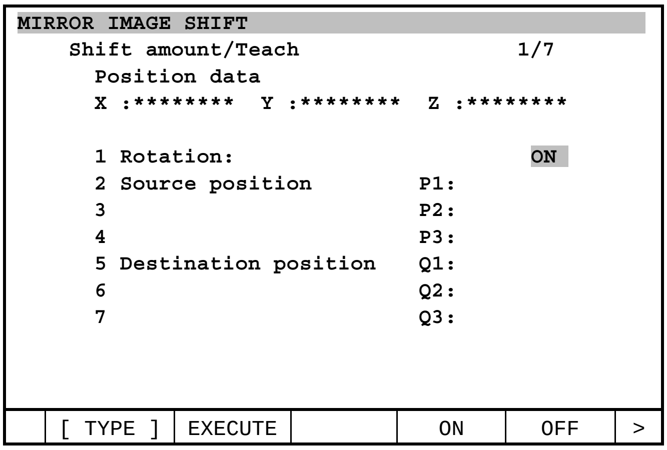
```
     Input position register number:

  [ TYPE ]  EXECUTE            REFER    RECORD   >
```

- **To clear a position setting,** press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

e. Record or specify the destination position.

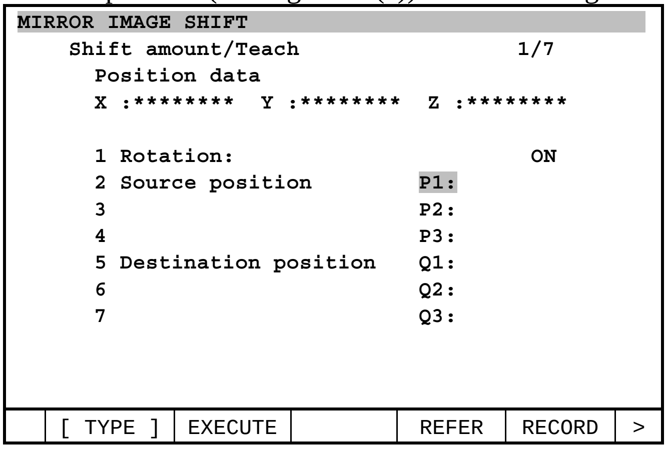
13. If you want to rotate the positions,

a. Press F4, ON. The following screen will be displayed.



```
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     >
```

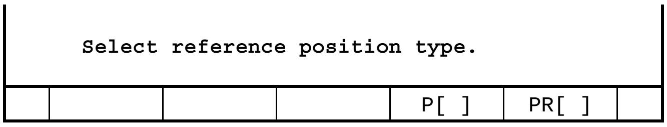
b. Move the cursor to source position (see Fig. 24.1 (a)). The following screen will be displayed.



```
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          REFER   RECORD    >
```

c. Move the robot to the first source position (P1) and either record or specify the position:

- **To record a position,** jog the robot to the position you want, press and hold in [SHIFT] key and press F5, RECORD.
- **To specify a previously recorded position or position register,** press F4, REFER. Select the position or position register.

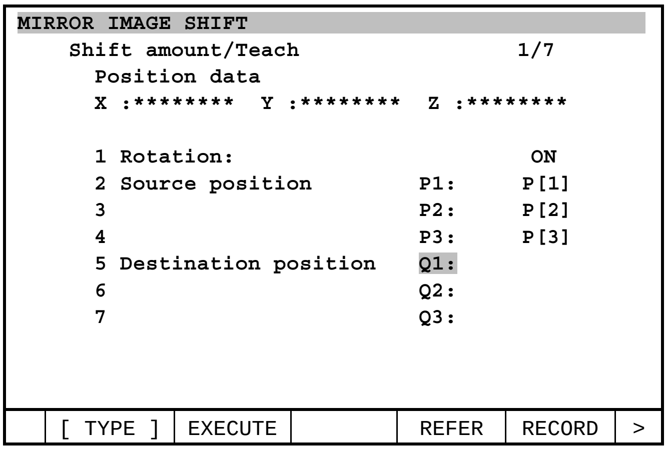


```
     Select reference position type.

                                  P[ ]    PR[ ]
```

- **To clear a position setting,** press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

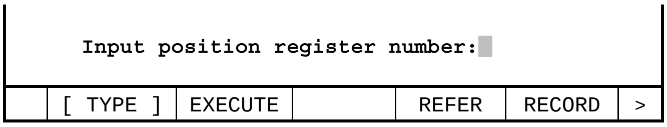
d. Record or specify all the source positions.

e. Move the cursor to Destination position. The following screen will be displayed.



```
MIRROR IMAGE SHIFT
    Shift amount/Teach                1/7
      Position data
      X :********  Y :********  Z :********

      1 Rotation:                      ON
      2 Source position        P1:     P[1]
      3                        P2:     P[2]
      4                        P3:     P[3]
      5 Destination position   Q1:
      6                        Q2:
      7                        Q3:

   [ TYPE ]  EXECUTE          REFER    RECORD   >
```

- **To record a position,** jog the robot to the destination position (Q1). Press and hold in the SHIFT key and press F5, RECORD.
- **To specify a previously recorded position register,** press F4, REFER. Type the number of the position register and press ENTER.



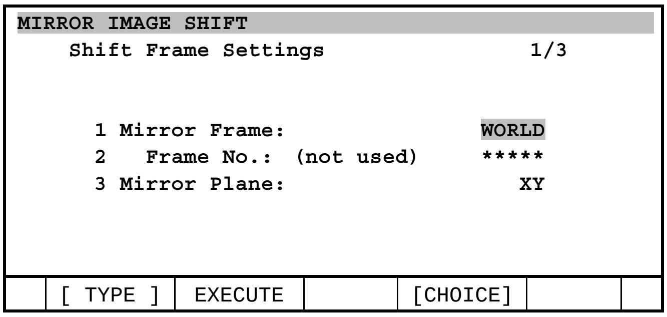
```
     Input position register number:

   [ TYPE ]  EXECUTE          REFER    RECORD   >
```

- **To clear a position setting,** press NEXT, >, and then press F1, CLEAR. At the prompt "Clear all data," press F4, YES, to confirm.

f. Record or specify all the destination positions.

⚠ **NOTE**
Pressing F2, EXECUTE, will cause the positions you have selected to be mirrored and **will not** cause robot motion.

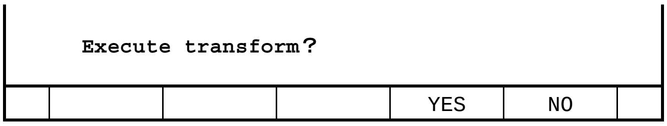
14. If you selected FRAME as the Image Shift Method, the following screen will be displayed.



```
MIRROR IMAGE SHIFT
    Shift Frame Settings               1/3


      1 Mirror Frame:                WORLD
      2   Frame No.:  (not used)     *****
      3 Mirror Plane:                   XY


   [ TYPE ]  EXECUTE         [CHOICE]
```

Move the cursor to Mirror Frame and press F4, [CHOICE]. Select the frame you want to use as the mirror frame and press ENTER.

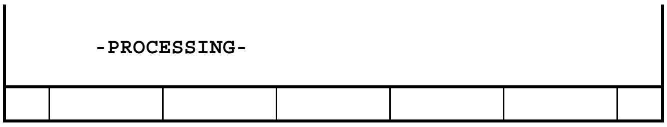
15. If you selected USER or JOG, move the cursor to Frame No. and type the number of the existing frame you want to use.
16. Move the cursor to Mirror Plane and press F4, [CHOICE]. Select the plane you want to use and press ENTER.
17. **To mirror image the program,** press F2, EXECUTE.



```
Execute transform?
```

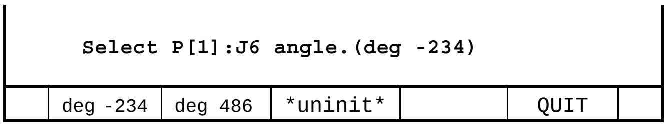
- **To execute the mirror image shift,** press F4, YES.
- **If you do not want to execute the mirror image shift,** press F5, NO.

18. Wait until software has finished processing the mirror image.



```
-PROCESSING-
```

If the turn number will be changed, you will see a message similar to the following.



```
Select P[1]:J6 angle.(deg -234)
```

The above example will be displayed when J6 turn number will be changed.

- If you do not want to change the turn number, press F1, deg –234.
- If you want to change the turn number, press F2, deg 486.
- If you want to make the position as un-initialized, press F3, *uninit*.
- If you want to interrupt mirror image, press F4, QUIT.

⚠ **CAUTION**

Do not attempt to move the robot to a position that was not mirrored correctly; otherwise, you could injure personnel or damage equipment.

When the mirror image is complete, you must reteach each position that did not mirror correctly.

25 CUSTOMIZE SUPPORT FUNCTION

In 7DC1 and 7DD0 software series Customize Support Function (J742) is included in KAREL Use Support Function.

This function supplies the necessary function for the one who develops the system specific function.
It supports following.

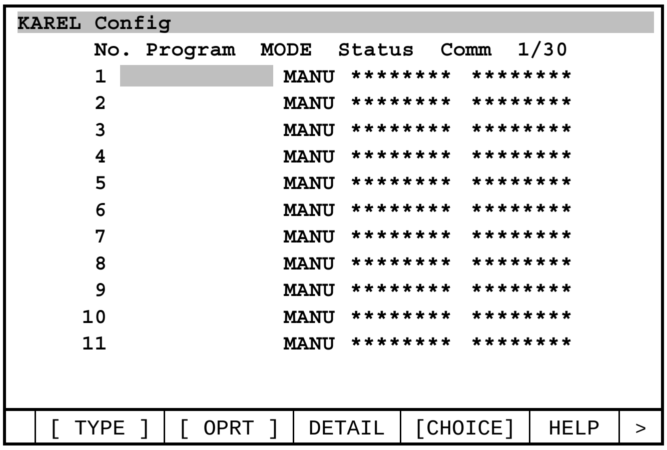
- Executing or aborting KAREL programs
- Auto running the KAREL programs at the time of power on
- Periodically monitor and ouput the specified KAREL program status
- Set the configuration of custom menu setting (cf. Section 25.3)

25.1 KAREL CONFIG

25.1.1 KAREL Config Screen

STEP

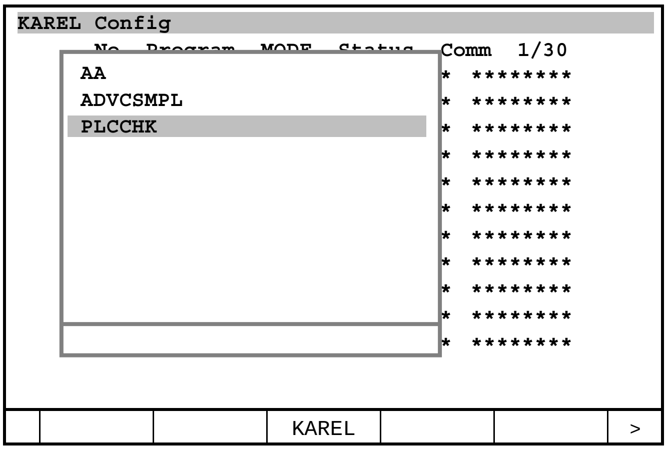
1 Select MENU – 6. SETUP – KAREL Config.
2 The following screen will be displayed.



```
KAREL Config
      No. Program  MODE  Status   Comm  1/30
       1             MANU ********  ********
       2             MANU ********  ********
       3             MANU ********  ********
       4             MANU ********  ********
       5             MANU ********  ********
       6             MANU ********  ********
       7             MANU ********  ********
       8             MANU ********  ********
       9             MANU ********  ********
      10             MANU ********  ********
      11             MANU ********  ********
```

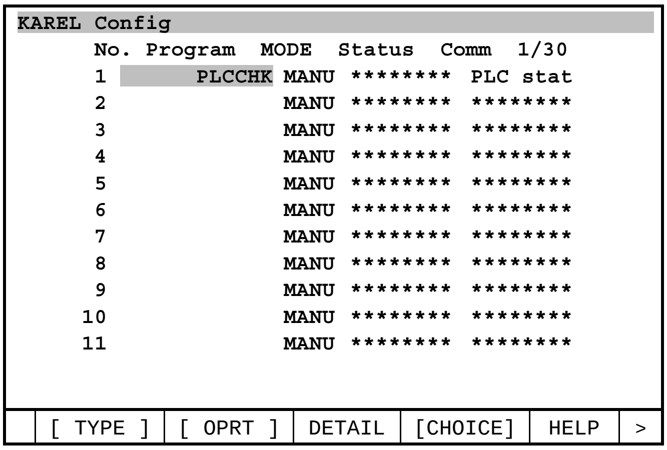
25.1.2 Use KAREL Config Screen

Select F4, [CHOICE], then you can see the list of KAREL program to be registered to run.



```
KAREL Config
                               Comm  1/30
    AA                         *  ********
    ADVCSMPL                   *  ********
    PLCCHK                     *  ********
                               *  ********
                               *  ********
                               *  ********
                               *  ********
                               *  ********
                               *  ********
                               *  ********
                               *  ********
```

In this documentation, PLCCHK is used as the sample to explain. Set the cursor onto the PLCCHK, and press [ENTER] key.



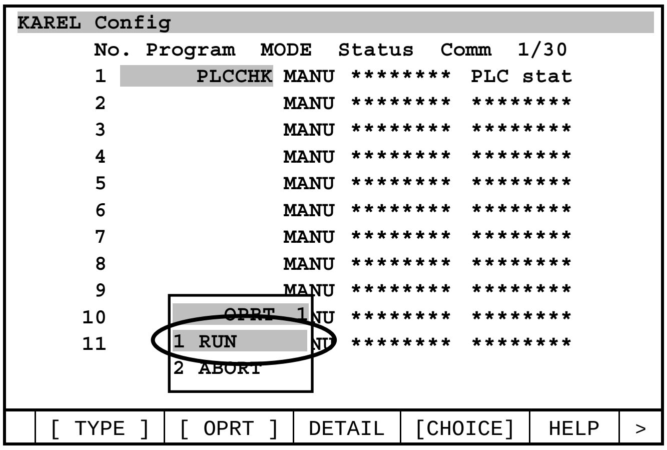
```
KAREL Config
      No. Program  MODE  Status  Comm  1/30
       1     PLCCHK MANU ******** PLC stat
       2            MANU ******** ********
       3            MANU ******** ********
       4            MANU ******** ********
       5            MANU ******** ********
       6            MANU ******** ********
       7            MANU ******** ********
       8            MANU ******** ********
       9            MANU ******** ********
      10            MANU ******** ********
      11            MANU ******** ********

  [ TYPE ] [ OPRT ] DETAIL [CHOICE] HELP >
```

There are four status, running, paused, aborted and ********. ******** means that status is not other 3 ones.

Comm is the comment that is described in the KAREL source file as %COMMENT.

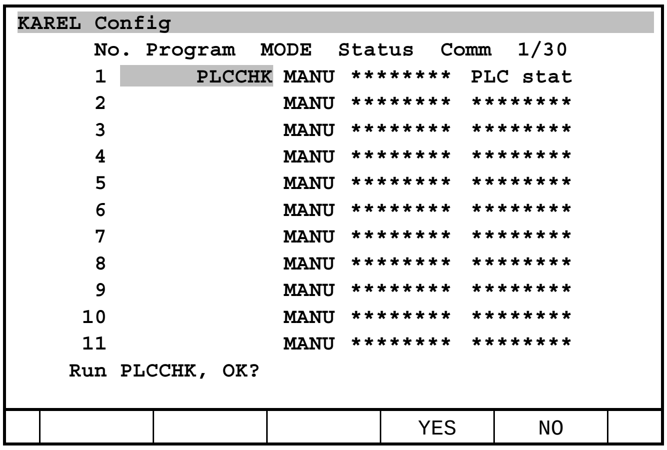
25.1.3 Run KAREL Program

Set the cursor onto PLCCHK in Program row and press F2, [OPRT].



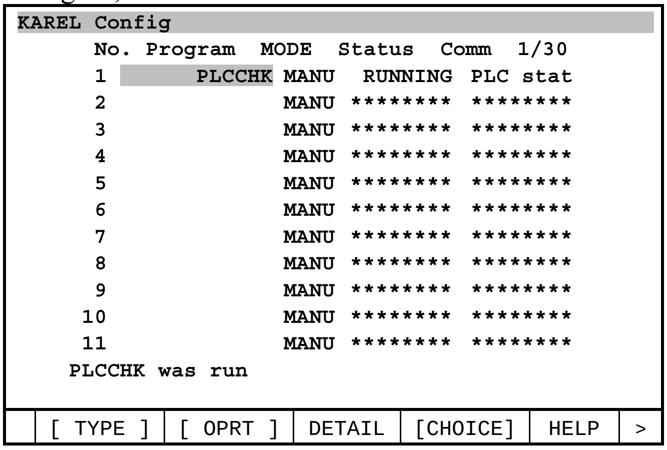
```
KAREL Config
      No. Program  MODE  Status  Comm  1/30
       1     PLCCHK MANU ******** PLC stat
       2            MANU ******** ********
       3            MANU ******** ********
       4            MANU ******** ********
       5            MANU ******** ********
       6            MANU ******** ********
       7            MANU ******** ********
       8            MANU ******** ********
       9            MANU ******** ********
      10     OPRT  1NU ******** ********
      11  1 RUN      NU ******** ********
          2 ABORT

  [ TYPE ] [ OPRT ] DETAIL [CHOICE] HELP >
```

Select 1. RUN. Then confirmation message is displayed.



```
KAREL Config
      No. Program  MODE  Status  Comm  1/30
       1     PLCCHK MANU ******** PLC stat
       2            MANU ******** ********
       3            MANU ******** ********
       4            MANU ******** ********
       5            MANU ******** ********
       6            MANU ******** ********
       7            MANU ******** ********
       8            MANU ******** ********
       9            MANU ******** ********
      10            MANU ******** ********
      11            MANU ******** ********
    Run PLCCHK, OK?

                          YES     NO
```

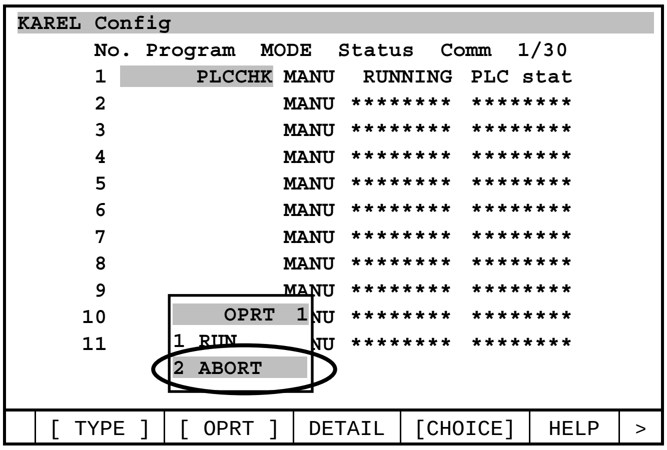
Program is run after selecting F4, YES.



```
KAREL Config
      No. Program  MODE  Status   Comm  1/30
       1       PLCCHK MANU  RUNNING  PLC stat
       2              MANU ******** ********
       3              MANU ******** ********
       4              MANU ******** ********
       5              MANU ******** ********
       6              MANU ******** ********
       7              MANU ******** ********
       8              MANU ******** ********
       9              MANU ******** ********
      10              MANU ******** ********
      11              MANU ******** ********
    PLCCHK was run

  [ TYPE ] [ OPRT ] DETAIL [CHOICE] HELP >
```

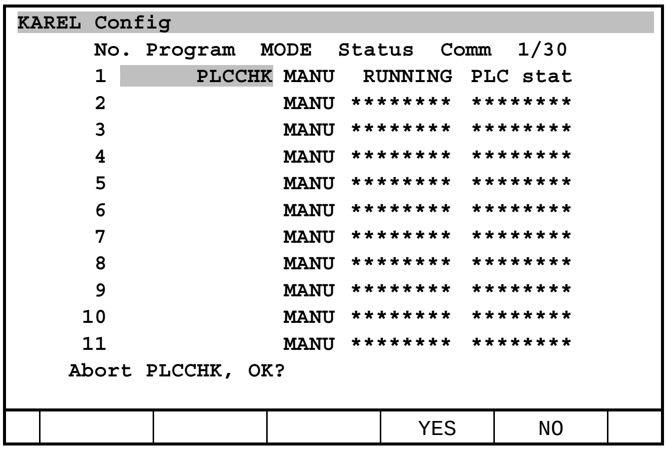
25.1.4 Abort KAREL Program

Set the cursor onto PLCCHK in Program row and press F2, [OPRT].



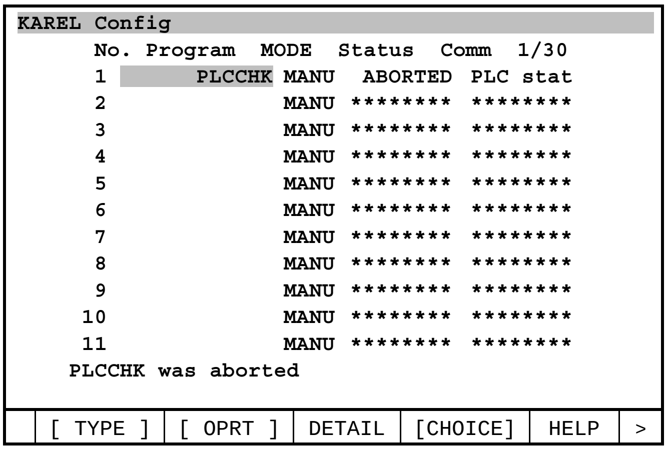
```
KAREL Config
      No. Program  MODE  Status   Comm  1/30
       1       PLCCHK MANU  RUNNING  PLC stat
       2              MANU ******** ********
       3              MANU ******** ********
       4              MANU ******** ********
       5              MANU ******** ********
       6              MANU ******** ********
       7              MANU ******** ********
       8              MANU ******** ********
       9              MANU ******** ********
      10          OPRT 1NU ******** ********
      11     1 RUN      NU ******** ********
             2 ABORT

  [ TYPE ] [ OPRT ] DETAIL [CHOICE] HELP >
```

Set the cursor onto 2. ABORT and press [ENTER] key.
Confirmation message will be displayed.



```
KAREL Config
      No. Program  MODE  Status   Comm  1/30
       1       PLCCHK MANU  RUNNING  PLC stat
       2              MANU ******** ********
       3              MANU ******** ********
       4              MANU ******** ********
       5              MANU ******** ********
       6              MANU ******** ********
       7              MANU ******** ********
       8              MANU ******** ********
       9              MANU ******** ********
      10              MANU ******** ********
      11              MANU ******** ********
    Abort PLCCHK, OK?

                              YES      NO
```

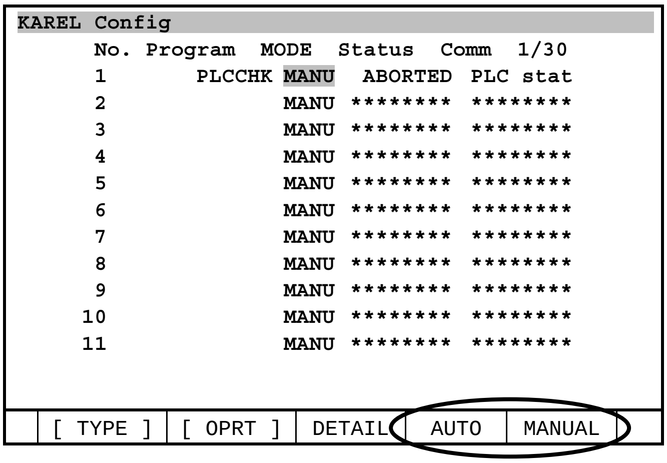
Select F4, YES, then KAREL program will be aborted.



```
KAREL Config
      No. Program  MODE  Status  Comm  1/30
       1     PLCCHK MANU  ABORTED  PLC stat
       2            MANU ********  ********
       3            MANU ********  ********
       4            MANU ********  ********
       5            MANU ********  ********
       6            MANU ********  ********
       7            MANU ********  ********
       8            MANU ********  ********
       9            MANU ********  ********
      10            MANU ********  ********
      11            MANU ********  ********
    PLCCHK was aborted

 [ TYPE ] [ OPRT ] DETAIL [CHOICE] HELP >
```

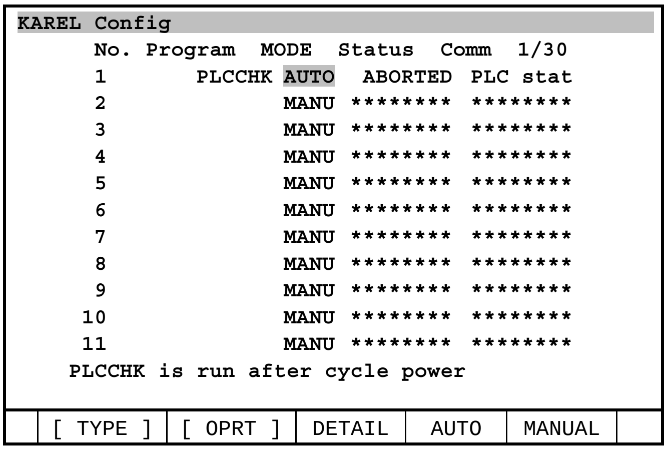
25.1.5 Start Mode Config of KAREL program

Set the cursor onto MODE row. F4 and F5 keys are turns to AUTO and MANUAL.



```
KAREL Config
      No. Program  MODE  Status  Comm  1/30
       1     PLCCHK MANU  ABORTED  PLC stat
       2            MANU ********  ********
       3            MANU ********  ********
       4            MANU ********  ********
       5            MANU ********  ********
       6            MANU ********  ********
       7            MANU ********  ********
       8            MANU ********  ********
       9            MANU ********  ********
      10            MANU ********  ********
      11            MANU ********  ********

 [ TYPE ] [ OPRT ] DETAIL  AUTO  MANUAL
```

Default is MANUAL. If you select AUTO, KAREL program is run after cycling power of the controller.

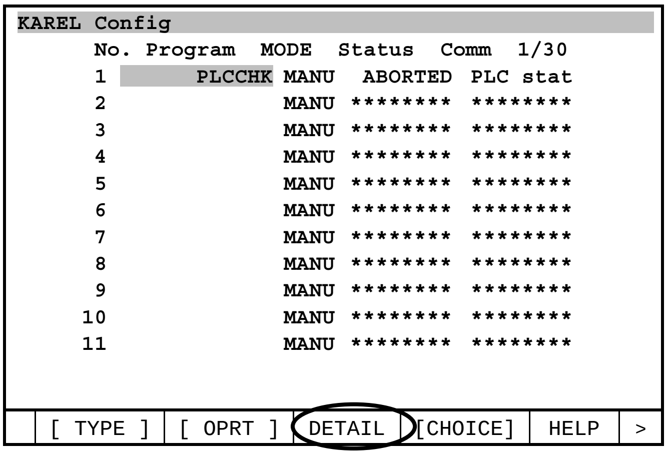


```
KAREL Config
      No. Program  MODE  Status  Comm  1/30
       1     PLCCHK AUTO  ABORTED  PLC stat
       2            MANU ********  ********
       3            MANU ********  ********
       4            MANU ********  ********
       5            MANU ********  ********
       6            MANU ********  ********
       7            MANU ********  ********
       8            MANU ********  ********
       9            MANU ********  ********
      10            MANU ********  ********
      11            MANU ********  ********
    PLCCHK is run after cycle power

 [ TYPE ] [ OPRT ] DETAIL  AUTO  MANUAL
```

There is a limit that the number of running the KAREL simultaneously. It depends on the configuration of the robot system. “PROG-014 Max task number exceed” will be posted when the number of running KAREL is going to be over max number.

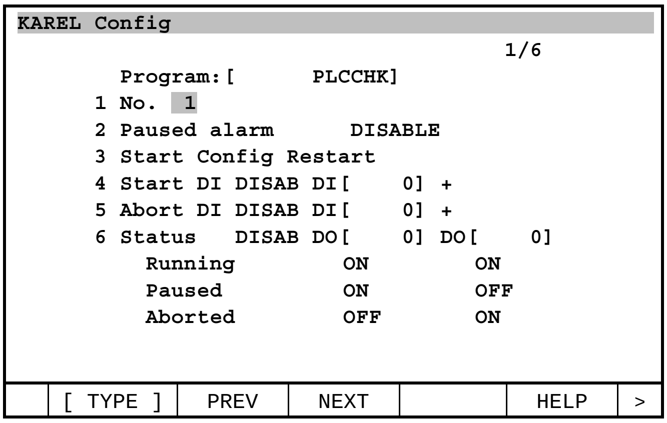
25.1.6 Detail Screen of KAREL Config

Press F3, DETAIL.



```
KAREL Config
      No. Program  MODE  Status  Comm  1/30
      1       PLCCHK MANU  ABORTED  PLC stat
      2              MANU ********  ********
      3              MANU ********  ********
      4              MANU ********  ********
      5              MANU ********  ********
      6              MANU ********  ********
      7              MANU ********  ********
      8              MANU ********  ********
      9              MANU ********  ********
     10              MANU ********  ********
     11              MANU ********  ********

  [ TYPE ] [ OPRT ]  DETAIL  [CHOICE]  HELP  >
```

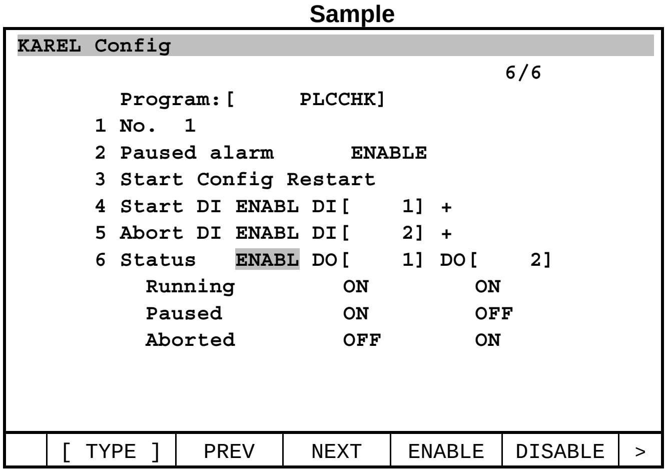
Like following screen will be displayed.



```
KAREL Config
                                        1/6
        Program:[      PLCCHK]
      1 No.  1
      2 Paused alarm      DISABLE
      3 Start Config Restart
      4 Start DI DISAB DI[    0] +
      5 Abort DI DISAB DI[    0] +
      6 Status   DISAB DO[    0] DO[    0]
          Running        ON        ON
          Paused         ON        OFF
          Aborted        OFF       ON

   [ TYPE ]   PREV    NEXT          HELP   >
```

Item	Comment	
Paused alarm	If ENABLE, warning is posted when specified program, ex PLCCHK, is paused or aborted.	
	Paused	"SYST 294 Paused (program name)" "SYST 296 Specified KAREL was paused"
	Aborted	"SYST 295 Aborted (program name)" "SYST 297 Specified KAREL was aborted"
Start Config	Sets the mode of restart.	
	Restart	Once aborting the specified program and start from the top of the program.
	Resume	Specified program is restarted from the paused point.
Start DI	Sets the start operation. Detecting the rising edge (+) or falling edge (-) of specified DI starts the specified program.	
	When "Start DI" is set enable, DI port number and Up or Down can not be changed. If you would like to set them, please set "Start DI" disable once. After setting them, please set it enable. Then this configuration is enabled after cycle power.	
Abort DI	Sets the start operation. Detecting the rising edge (+) or falling edge (-) of specified DI starts the specified program. When "Start DI" is set enable, DI port number and Up or Down can not be changed. If you would like to set them, please set "Start DI" disable once. After setting them, please set it enable. Then this configuration is enabled after cycle power.	
Status	Is to monitor the specified program and output its status with two DOs. As the default, program is monitored every 100msec. Its config is enabled after cycle power.	

Sample



```
KAREL Config
                                           6/6
        Program:[     PLCCHK]
      1 No.  1
      2 Paused alarm      ENABLE
      3 Start Config Restart
      4 Start DI ENABL DI[    1] +
      5 Abort DI ENABL DI[    2] +
      6 Status   ENABL DO[    1] DO[    2]
          Running        ON        ON
          Paused         ON        OFF
          Aborted        OFF       ON
```

PLCCHK status	DO[1]	DO[2]
Running	ON	ON
Paused	ON	OFF
Aborted	OFF	ON

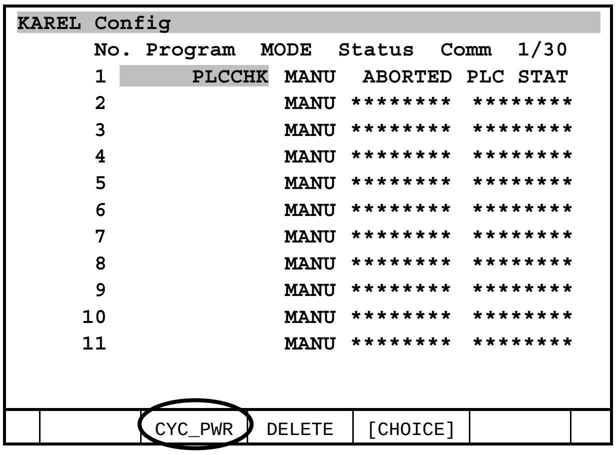
To switch previous screen, press F2, PREV. In this case, screen is switched to No.30.
To switch next detail screen, press F3, NEXT. In this case, screen is switched to No.2.

25.1.7 Limitation and Caution of KAREL Config

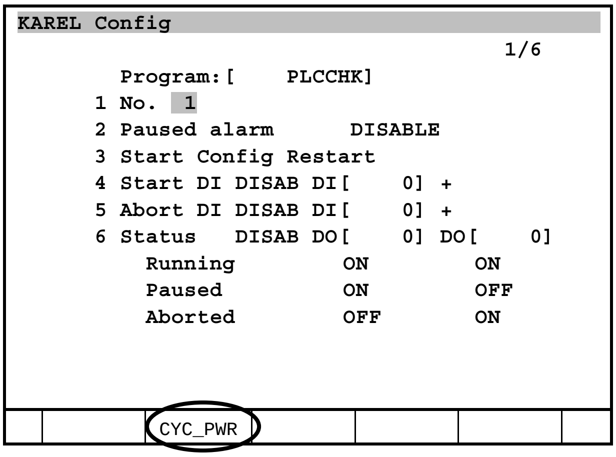
- Status is monitored at a constant frequency. Thus status is not changed at a real time. For example, program status is changed from running to paused, and aborted in one monitoring cycle, paused status is not detected at next monitoring cycle. Therefore status is changed from running to aborted in this case.
- Monitoring constant frequency can be changed. You can change it with system variable $KRL_NUM.$UPDT_TIME. It is able to change in the range 8 to 5000 msec. If you set it out of its range under 8 or above 5000, it is automatically set to 8 or 5000msec.
- In the case of above sample, after you set the detail of PLCCHK, return to KAREL Config screen with [PREV] key and change the registered No.1 program from PLCCHK to other program, such as C2, PLCCHK does not start if DI[1] is turned on. C2 is run instead. For prevention of miss operation, please do not change program after set the detail.

25.1.8 Cycle Power (R-30*i*B Controller)

Both KAREL config and detail screen enable to cycle power.
In R-30*i*B Mate controller, do cycle power manually.

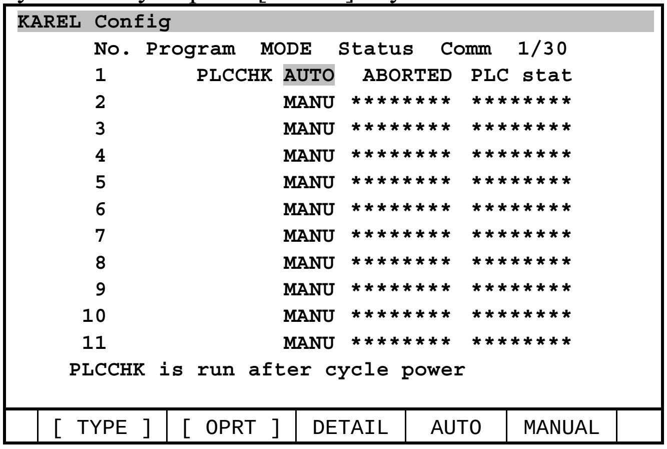


```
KAREL Config
      No. Program  MODE  Status  Comm  1/30
      1       PLCCHK MANU  ABORTED PLC STAT
      2              MANU ********  ********
      3              MANU ********  ********
      4              MANU ********  ********
      5              MANU ********  ********
      6              MANU ********  ********
      7              MANU ********  ********
      8              MANU ********  ********
      9              MANU ********  ********
     10              MANU ********  ********
     11              MANU ********  ********

            CYC_PWR  DELETE  [CHOICE]
```



```
KAREL Config
                                             1/6
        Program:[    PLCCHK]
      1 No.  1
      2 Paused alarm      DISABLE
      3 Start Config Restart
      4 Start DI DISAB DI[    0] +
      5 Abort DI DISAB DI[    0] +
      6 Status   DISAB DO[    0] DO[    0]
          Running         ON         ON
          Paused          ON         OFF
          Aborted         OFF        ON

            CYC_PWR
```

F2, CYC_PWR is displayed when you press [NEXT] key. Set the TP to enable and press F2, CYC_PWR.



```
KAREL Config
      No. Program  MODE  Status  Comm  1/30
      1       PLCCHK AUTO  ABORTED  PLC stat
      2              MANU ********  ********
      3              MANU ********  ********
      4              MANU ********  ********
      5              MANU ********  ********
      6              MANU ********  ********
      7              MANU ********  ********
      8              MANU ********  ********
      9              MANU ********  ********
     10              MANU ********  ********
     11              MANU ********  ********
    PLCCHK is run after cycle power

  [ TYPE ] [ OPRT ]  DETAIL   AUTO   MANUAL
```

Selecting F4, YES do cycle power.

25.2 CUSTOM MENU

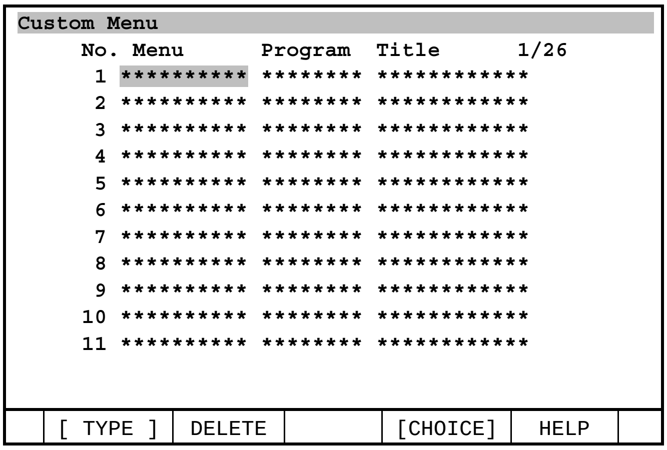
25.2.1 Overview

This function allows to set the TP or KAREL program to the menu. It is easy to run the specifed program to select the menu.

For example, user-created screen by KAREL program is allowed to display quickly by selecting the menu.

25.2.2 Starting Custom Menu

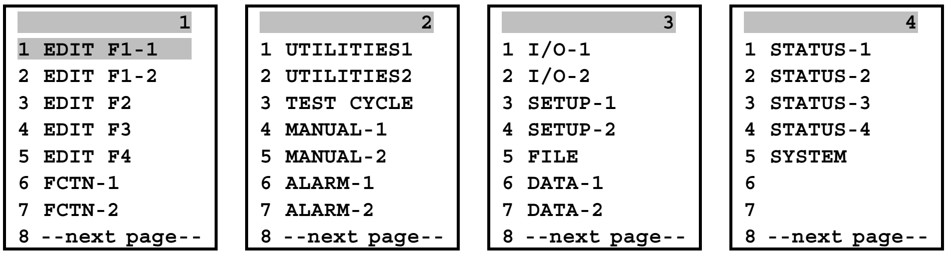
Select MENU – setup – Custom Menu.
The following screen will be displayed.



```
Custom Menu
     No. Menu       Program  Title      1/26
       1 ********** ******** ************
       2 ********** ******** ************
       3 ********** ******** ************
       4 ********** ******** ************
       5 ********** ******** ************
       6 ********** ******** ************
       7 ********** ******** ************
       8 ********** ******** ************
       9 ********** ******** ************
      10 ********** ******** ************
      11 ********** ******** ************

  [ TYPE ]  DELETE            [CHOICE]   HELP
```

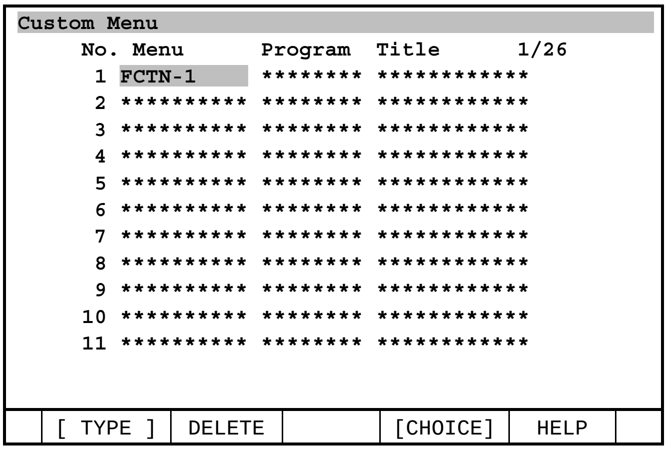
25.2.3 Set Custom Menu

Press F4, [CHOICE].



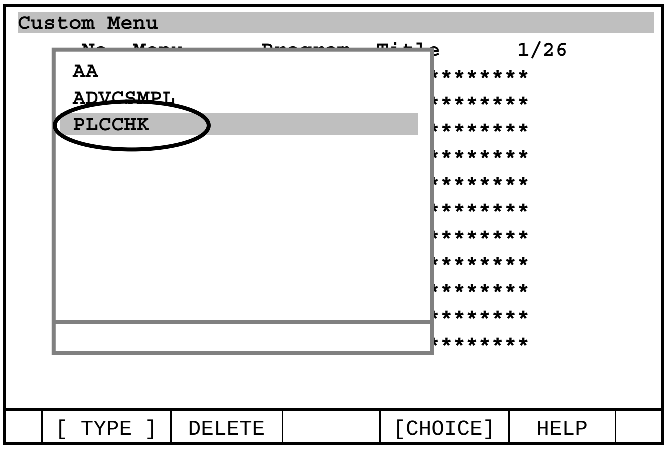
```
              1                  2                  3                  4
1 EDIT F1-1        1 UTILITIES1       1 I/O-1            1 STATUS-1
2 EDIT F1-2        2 UTILITIES2       2 I/O-2            2 STATUS-2
3 EDIT F2          3 TEST CYCLE       3 SETUP-1          3 STATUS-3
4 EDIT F3          4 MANUAL-1         4 SETUP-2          4 STATUS-4
5 EDIT F4          5 MANUAL-2         5 FILE             5 SYSTEM
6 FCTN-1           6 ALARM-1          6 DATA-1           6
7 FCTN-2           7 ALARM-2          7 DATA-2           7
8 --next page--    8 --next page--    8 --next page--    8 --next page--
```

Select item that you would like to add. Select FCTN –1 as an example.



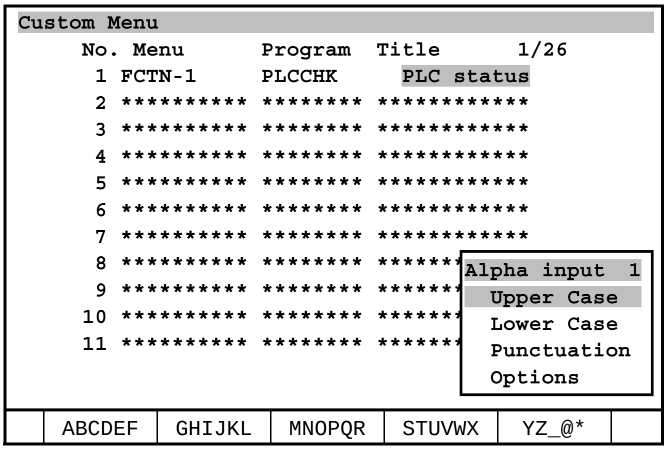
```
Custom Menu
     No. Menu       Program  Title      1/26
       1 FCTN-1     ******** ************
       2 ********** ******** ************
       3 ********** ******** ************
       4 ********** ******** ************
       5 ********** ******** ************
       6 ********** ******** ************
       7 ********** ******** ************
       8 ********** ******** ************
       9 ********** ******** ************
      10 ********** ******** ************
      11 ********** ******** ************

  [ TYPE ]  DELETE            [CHOICE]   HELP
```

Set the cursor onto program row and press F4, [CHOICE] key. As an example PLCCHK is selected.



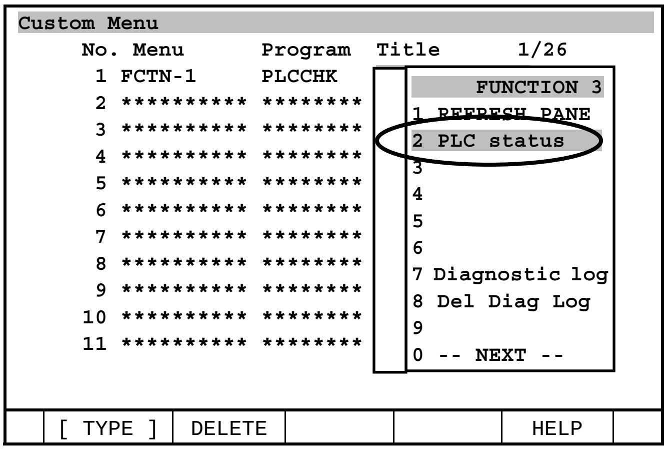
```
Custom Menu
                                          1/26
        AA                       ********
        ADVCSMPL                 ********
        PLCCHK                   ********
                                 ********
                                 ********
                                 ********
                                 ********
                                 ********
                                 ********
                                 ********
                                 ********
 [ TYPE ]  DELETE        [CHOICE]  HELP
```

Next, set the cursor onto title row and press [ENTER] key. PLCCHK is input for instance.



```
Custom Menu
     No. Menu       Program  Title       1/26
      1 FCTN-1      PLCCHK     PLC status
      2 ********** ******** ************
      3 ********** ******** ************
      4 ********** ******** ************
      5 ********** ******** ************
      6 ********** ******** ************
      7 ********** ******** ************
      8 ********** ******** ******Alpha input  1
      9 ********** ******** ******  Upper Case
     10 ********** ******** ******  Lower Case
     11 ********** ******** ******  Punctuation
                                    Options
  ABCDEF  GHIJKL  MNOPQR  STUVWX  YZ_@*
```

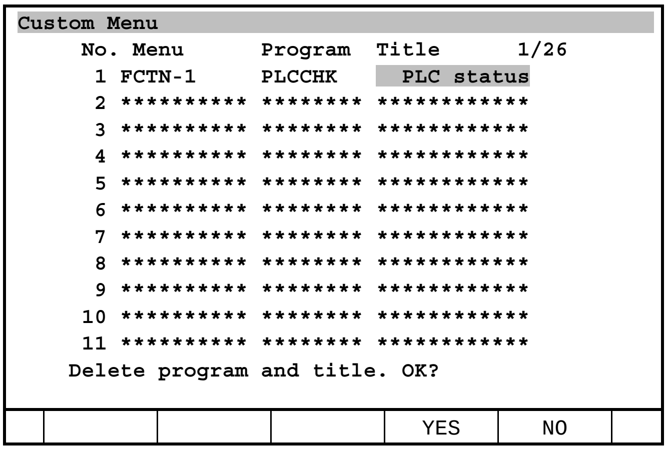
Select FCTN → 0 -- NEXT -- → 0 --NEXT --. Verify PLCCHK is added to FCTN menu.



```
Custom Menu
     No. Menu       Program  Title       1/26
      1 FCTN-1      PLCCHK
      2 ********** ********     FUNCTION 3
      3 ********** ********  1 REFRESH PANE
      4 ********** ********  2 PLC status
      5 ********** ********  3
      6 ********** ********  4
      7 ********** ********  5
      8 ********** ********  6
      9 ********** ********  7 Diagnostic log
     10 ********** ********  8 Del Diag Log
     11 ********** ********  9
                             0 -- NEXT --
 [ TYPE ]  DELETE                  HELP
```

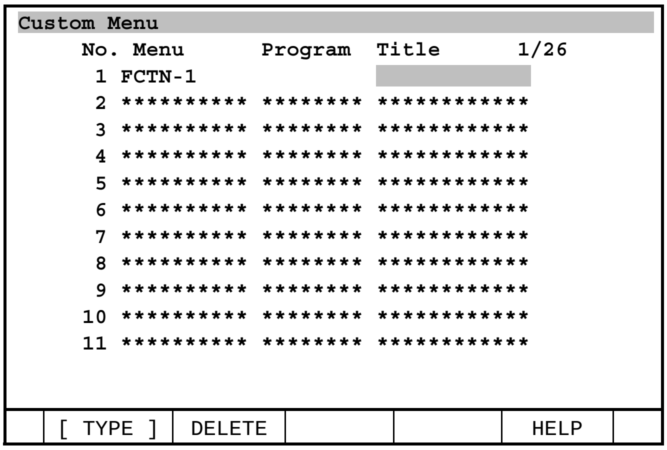
25.2.4 Delete Set

In the case of deleting set, press F2, DELETE. Confirmation message will be displayed like following.



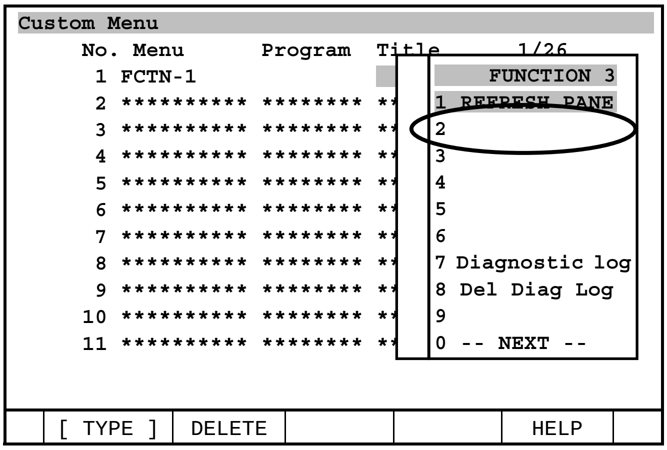
```
Custom Menu
     No. Menu       Program  Title       1/26
       1 FCTN-1     PLCCHK     PLC status
       2 ********** ******** ************
       3 ********** ******** ************
       4 ********** ******** ************
       5 ********** ******** ************
       6 ********** ******** ************
       7 ********** ******** ************
       8 ********** ******** ************
       9 ********** ******** ************
      10 ********** ******** ************
      11 ********** ******** ************
    Delete program and title. OK?
```

Select F4, YES.
Specified program and title will be deleted.



```
Custom Menu
     No. Menu       Program  Title       1/26
       1 FCTN-1
       2 ********** ******** ************
       3 ********** ******** ************
       4 ********** ******** ************
       5 ********** ******** ************
       6 ********** ******** ************
       7 ********** ******** ************
       8 ********** ******** ************
       9 ********** ******** ************
      10 ********** ******** ************
      11 ********** ******** ************
```

Select FCTN → 0 -- NEXT -- → 0 -- NEXT --. Verify PLCCHK will be deleted from FCTN menu.



26 KAREL PROGRAM EXECUTION HISTORY RECORD

KAREL program execution history record is a tool used to diagnose user application or controller system software problems. This option generates logs that show details about events, including the sequence in which they occur.

This option allows you to log certain events that occur during KAREL and TPP program execution. It also allows you to generate ASCII files that contain all or part of the logged data. The ASCII files contain detailed information about each event that is logged.

For example, you can log the following types of events.

- Start of execution of KAREL statements or TPP lines.
- Routine calls or returns.
- Internal system events for diagnosis of system software problems.

Refer to the following for additional information.

- For information about related hardware and software, refer to 26.1 HARDWARE AND SOFTWARE.
- For information about installing and setting up the debugging option, refer to 26.2 SETUP AND OPERATIONS.
- For information about events, refer to Chapter 26.3 LOGGING EVENTS.
- For examples of this option as related to KAREL programming, teach pendant programming, and the ASCII file format, refer to 26.4 EXAMPLES.

26.1 HARDWARE AND SOFTWARE

26.1.1 Hardware and Software Requirements

There are certain hardware and software requirements that must be met before you can run the KAREL Program Execution History Record.

NOTE
The hardware requirements in this section apply to all robot models.

26.1.2 Hardware

The KAREL Program Execution History Record requires the following hardware for all robot models:

- A FANUC Robot Controller
- A teach pendant
- 42000 bytes of Temporary memory
- 18000 bytes of CMOS
- 30000 bytes of FROM file system space

26.1.3 Software

The KAREL Program Execution History Recode function is included in KAREL Use Support function option (J971).

A series of teach pendant is used to control the logging of diagnostic information. The teach pendant screens allow you to:

- Specify the kinds of events you want to log.
- Specify the tasks for which events are to be logged.
- Generate ASCII files containing the logged data.

26.1.4 Performance

The KAREL Program Execution History Record Option can affect the performance of your system.

- This option reduces execution speed of KAREL and teach pendant program logic by approximately 1 to 2 %.

> **NOTE**
> You can select “Disable all logging” on the Diagnostic Logging main menu to eliminate this reduction.

Refer to Subsection 26.3.8 for more information about system performance.

- Actual logging of events adds about .08 ms per event logged.
- Motion execution is not affected.

26.2 SETUP AND OPERATIONS

26.2.1 Setting Up the KAREL Program Execution History Record

Before you can log information and write data log files, you must set up the KAREL Program Execution History Record Option. This section contains information about how to set up the KAREL Program Execution History Record Option, including a detailed procedure for each setup screen.

The following screens are used to control diagnostic logging:

- Clear event log deletes all data from the log.
- Dump Selections screen allows you to write log data to an ASCII file, Subsection 26.2.2.
- Task Selection screen allows you to add tasks for which events are logged, Subsection 26.2.3.
- Stop Logging a task Selection screen allows you to select tasks to be removed from the list of currently logged tasks, Subsection 26.2.4.
- Currently Selected tasks screen lists tasks currently selected for logging, Subsection 26.2.5.
- Event Class Selections screen allows you to select or de-select classes or groups of related event types for logging, Subsection 26.2.6.
- Event Detail selections screen allows you to select or de-select individual event types for logging, Subsection 26.2.7.
- Enable or Disable all logging allows you to start or stop all events logging, Subsection 26.2.8.

For example, you can log the following types of events.

- Start of execution of KAREL statements or TPP lines.
- Routine calls or returns.
- Internal system events for diagnosis of system software problems.

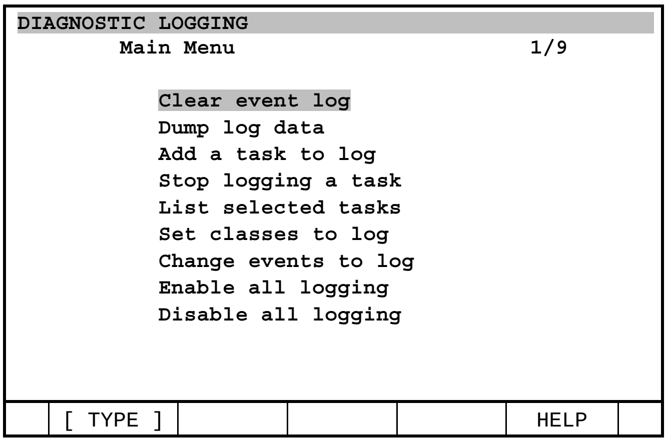
26.2.2 Dump Selections Screen

The Dump Selection Screen allows you to select a range of records to be dumped or logged. If the log is currently empty, an error message is displayed. Use Procedure 26-1 to dump records to the debug log.

Procedure 26-1 Dumping Records to the Debug Log

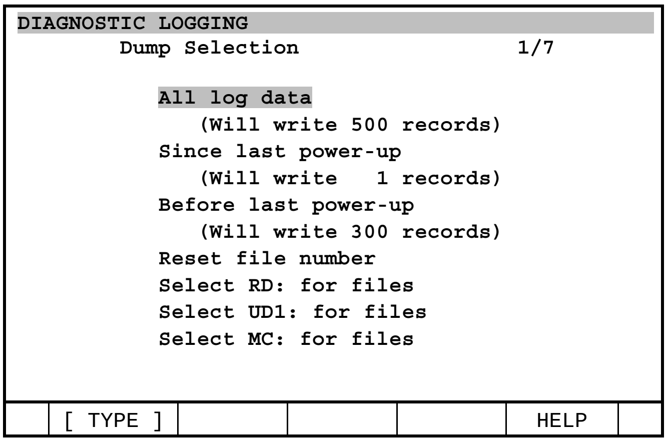
Steps

1. Press [MENU] key.
2. Select TEST CYCLE.
3. Press F1, [TYPE].
4. Select Debug Ctl, the following screen will be displayed.



```
DIAGNOSTIC LOGGING
        Main Menu                      1/9

           Clear event log
           Dump log data
           Add a task to log
           Stop logging a task
           List selected tasks
           Set classes to log
           Change events to log
           Enable all logging
           Disable all logging

   [ TYPE ]                          HELP
```

5. Select Dump log data. The number of records in each range is displayed on the screen. See the following screen for an example.



```
DIAGNOSTIC LOGGING
        Dump Selection                1/7

           All log data
              (Will write 500 records)
           Since last power-up
              (Will write   1 records)
           Before last power-up
              (Will write 300 records)
           Reset file number
           Select RD: for files
           Select UD1: for files
           Select MC: for files

   [ TYPE ]                          HELP
```

6. Select the range of data to be dumped or logged.
   - To request a dump of all event information currently in the log, select All log data.
   - To request a dump of all event information logged since the most recent power-up, select Since last power-up.
   - To request a dump of all event information between the last two power-ups, select Before last power-up. This will log all events prior to the last power-down.

   The information will be dumped to an ASCII file.

NOTE

By default, successive log files are written to MC:PGDBG201.DT, MC:PGDBG202.DT, and so forth.

7. To reset the file number in the file naming convention, select Reset file number. The starting point is MC:PGDBG201.DT.

8. To change the default saving device to RAM disk, select RD: for files. All files will be automatically saved to RAM disk.

9. To change the default saving device to USB memory, select UD1: for files. All files will automatically be saved to USB memory.

10. To change the default saving device to memory card, select MC: for files. All files will automatically be saved to a memory card.

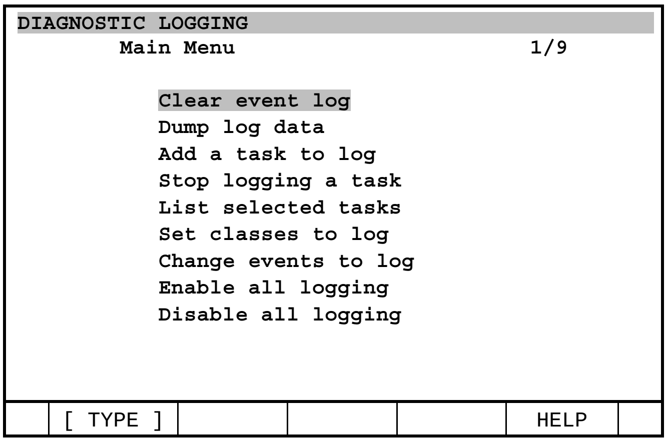
26.2.3 Task Selection Screen

The Task Selection screen allows you to select additional tasks for which you want to log information. Use Procedure 26-2 to select tasks to log.

Procedure 26-2 Selecting Tasks to Log

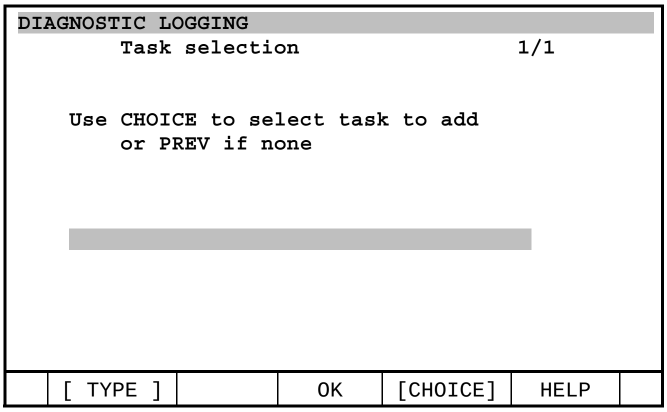
Steps

1 Press [MENU] key.
2 Select TEST CYCLE.
3 Press F1, [TYPE].
4 Select Debug Ctl. The following screen will be displayed.



```
DIAGNOSTIC LOGGING
        Main Menu                        1/9

           Clear event log
           Dump log data
           Add a task to log
           Stop logging a task
           List selected tasks
           Set classes to log
           Change events to log
           Enable all logging
           Disable all logging
```

5 Select Add a task to log. The following screen will be displayed.

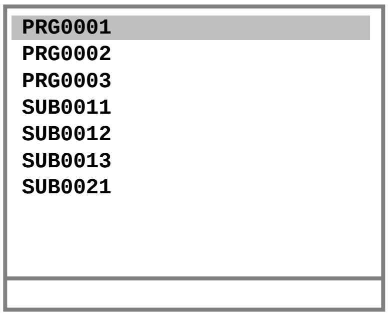


```
DIAGNOSTIC LOGGING
        Task selection                  1/1


    Use CHOICE to select task to add
        or PREV if none
```

6 To display a list of tasks from which you can select, press F4, [CHOICE]. You will see a screen similar to the following.

NOTE

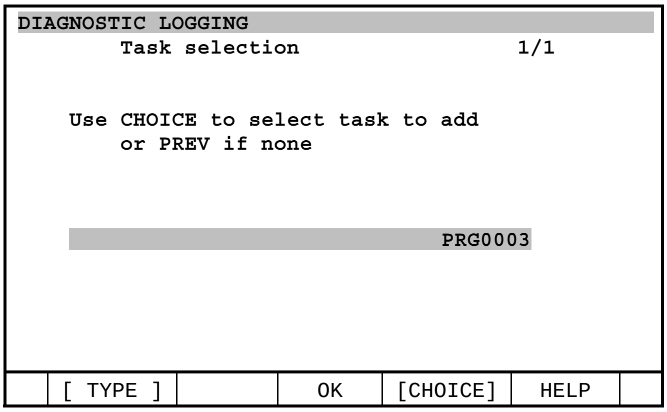
Selecting a program means that events will be logged for a task that has the selected program as its MAIN program.



```
PRG0001
PRG0002
PRG0003
SUB0011
SUB0012
SUB0013
SUB0021
```

7 Use the arrow keys to select the task for which you want events logged.

8 Press [ENTER] key when you have chosen a task. The following screen will be displayed.



```
DIAGNOSTIC LOGGING
        Task selection                1/1


    Use CHOICE to select task to add
        or PREV if none



                             PRG0003
```

9 To request logging for the displayed task, press F3, OK.

⚠CAUTION

Do not exit this screen by pressing PREV, F1, [TYPE], or any other key before you press F3, OK; otherwise, the selected task will not be logged.

26.2.4 Stop Logging Tasks Screen

The Stop Logging Tasks screen allows you to select tasks to be removed from the list of currently logged tasks. Use Procedure 26-3 to stop logging tasks.

Procedure 26-3 Stop Logging Tasks

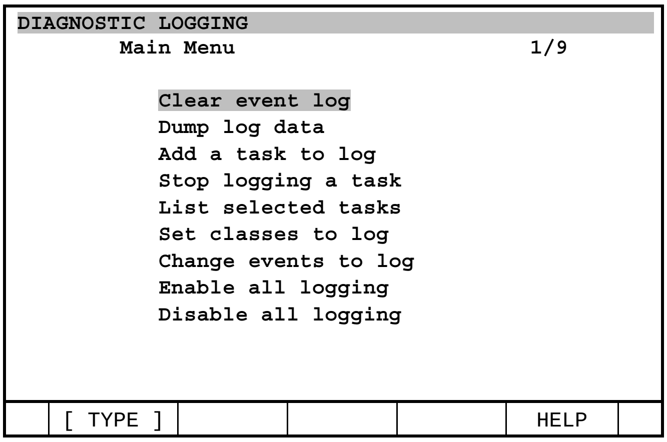
Steps

1 Press [MENU] key.

2 Select TEST CYCLE.

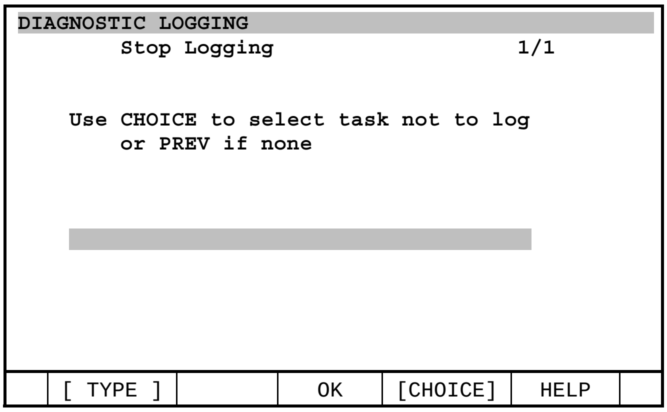
3 Press F1, [TYPE].

4 Select Debug Ctl. The following screen will be displayed.



```
DIAGNOSTIC LOGGING
        Main Menu                       1/9

            Clear event log
            Dump log data
            Add a task to log
            Stop logging a task
            List selected tasks
            Set classes to log
            Change events to log
            Enable all logging
            Disable all logging

   [ TYPE ]                          HELP
```

5 Select Stop logging a task. The following screen will be displayed.

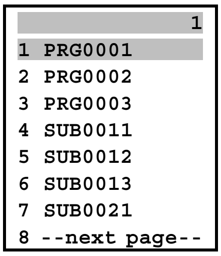


```
DIAGNOSTIC LOGGING
        Stop Logging                   1/1


    Use CHOICE to select task not to log
        or PREV if none



   [ TYPE ]          OK    [CHOICE]  HELP
```

6 To display a list of tasks, from which you can select, press F4, [CHOICE]. You will see a screen similar to the following.

NOTE

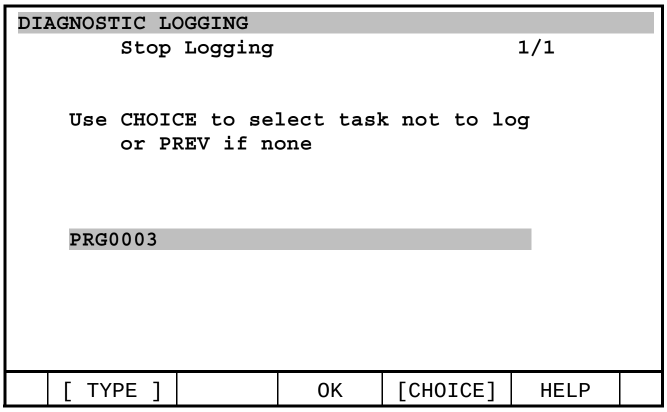
When you select a program, the events will no longer be logged for any tasks that have the selected program as their MAIN program.



```
             1
1 PRG0001
2 PRG0002
3 PRG0003
4 SUB0011
5 SUB0012
6 SUB0013
7 SUB0021
8 --next page--
```

7 Use the arrow keys to select the task for which you want to stop logging events.

8 Press ENTER when you have chosen a task. The following screen will be displayed.



```
DIAGNOSTIC LOGGING
        Stop Logging                   1/1


    Use CHOICE to select task not to log
        or PREV if none



    PRG0003
```

9 To stop logging for the task that is current displayed, press F3, OK.

CAUTION
Do not exit this screen by pressing PREV, F1, [TYPE] , or any other key before you press F3, OK; otherwise, logging for the selected task will not stop.

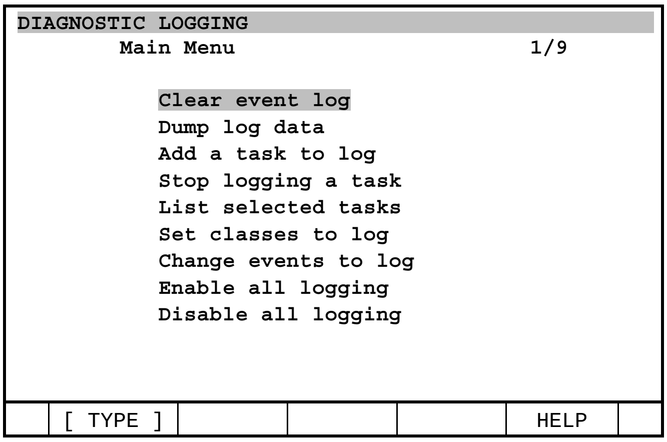
26.2.5 List Selected Tasks Screen

The List Selected Tasks Screen allows you to displays the tasks that are currently selected and being logged. Use Procedure 26-4 to list selected tasks.

Procedure26-4 List Selected Tasks

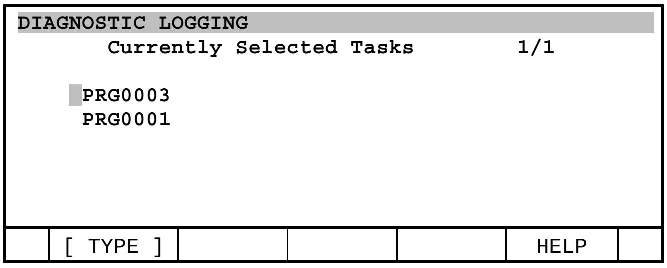
Steps

1 Press [MENU] key.
2 Select TEST CYCLE.
3 Press F1, [TYPE].
4 Select Debug Ctl. The following screen will be displayed.



```
DIAGNOSTIC LOGGING
        Main Menu                        1/9

           Clear event log
           Dump log data
           Add a task to log
           Stop logging a task
           List selected tasks
           Set classes to log
           Change events to log
           Enable all logging
           Disable all logging
```

5 Select List selected tasks. The following screen will be displayed.



```
DIAGNOSTIC LOGGING
       Currently Selected Tasks        1/1

    PRG0003
    PRG0001
```

This screen displays the tasks that are currently selected and being logged.

6 Press [PREV] key on the teach pendant to return to the main menu.

26.2.6 Event Class Selection Screen

The Event Class Selection Screen allows you to determine which classes of events should be logged (Procedure26-7). Each class of events enables or disables one or more detailed event types. The following is a list of event classes and the associated event types. Refer to Subsection 26.4.1 for more detailed information on the types of events that can be logged.

NOTE
An asterisk (*) indicates logging is for internal use only.

- Execute KAREL statement
  - Execute KAREL statement
- Execute TPP line
  - Execute TPP line
- Call/return
  - KAREL or TPP routine called
  - KAREL or TPP routine returned
- Motion *
  - Motion started *
  - Motion planned (teach pendant motion only) *
  - Motion cancel issued *
  - Motion stop issued *
  - Resume move *
  - Motion done received *
  - Motion completed normally *
  - MMR received *
- Condition Handler *
  - Condition handler triggered *
  - Condition handler enabled *
  - Condition handler disabled *
- Interrupt rtn
  - Before transferring to an ISR (Interrupt Sub-Routine)
  - After transferring to an ISR (Interrupt Sub-Routine)
  - Return from ISR (Interrupt Sub-Routine)
- Task start/end
  - KAREL or TPP task starts execution
  - KAREL or TPP task aborts
- Packet_rcd *
  - Interpreter receives packet *
- Pcode exec *
  - Starting execution of p-code *
- AMR activity *

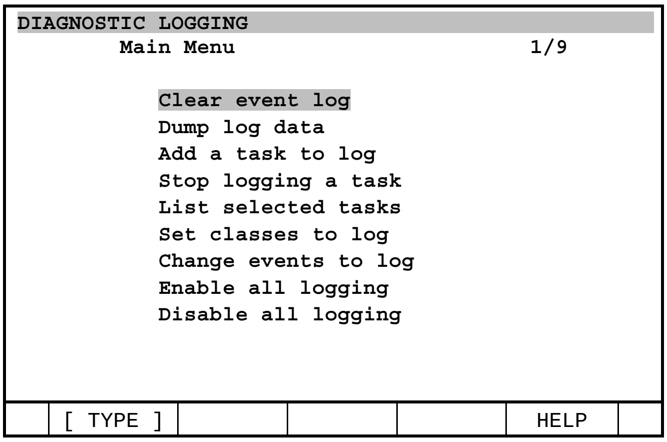
- Normal AMR recvd by AMGR *
- Start AMR recvd by AMGR *
- Stop AMR recvd by AMGR *
- AMR sent to AX *
- AMR rcvd from AX *

Use Procedure 26-5 to enable or disable event classes for logging.

Procedure 26-5 Enable or Disable Event Classes for Logging

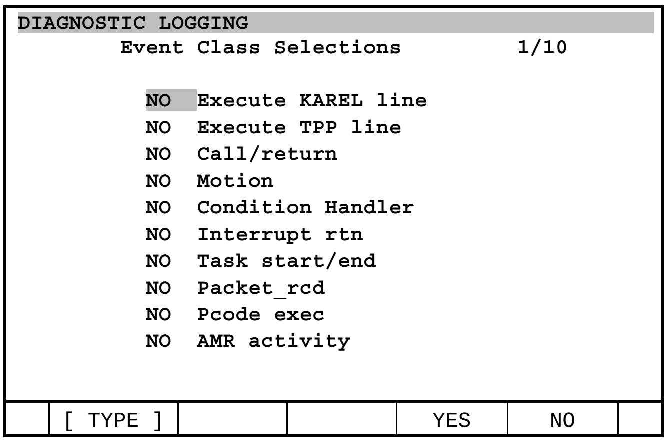
Steps

1 Press [MENU] key.
2 Select TEST CYCLE.
3 Press F1, [TYPE].
4 Select Debug Ctl. The following screen will be displayed.



```
DIAGNOSTIC LOGGING
        Main Menu                        1/9

           Clear event log
           Dump log data
           Add a task to log
           Stop logging a task
           List selected tasks
           Set classes to log
           Change events to log
           Enable all logging
           Disable all logging

   [ TYPE ]                           HELP
```

5 Select Set classes to log. The following screen will be displayed.



```
DIAGNOSTIC LOGGING
        Event Class Selections          1/10

          NO  Execute KAREL line
          NO  Execute TPP line
          NO  Call/return
          NO  Motion
          NO  Condition Handler
          NO  Interrupt rtn
          NO  Task start/end
          NO  Packet_rcd
          NO  Pcode exec
          NO  AMR activity

   [ TYPE ]                  YES        NO
```

6 Enable or disable classes of events to be logged.
- To turn on logging for a class of events, press F4, YES.
- To turn off logging for a class of events, press F5, NO.

The changes take effect immediately. The current status of whether an event is being logged is displayed to the left of the event.

NOTE
If you specify an entire class of events, you will enable or disable logging for all event types in that class.

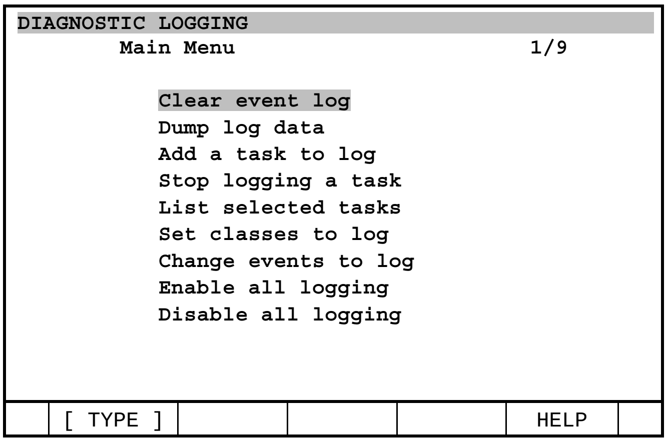
26.2.7 Event Detail Selection Screen

The Event Detail Selections Screen allows you to enable and disable logging of specific events. Use Procedure 26-6 to enable or disable a specific event type for logging.

Procedure 26-6 Enable or Disable a Specific Event Type for Logging

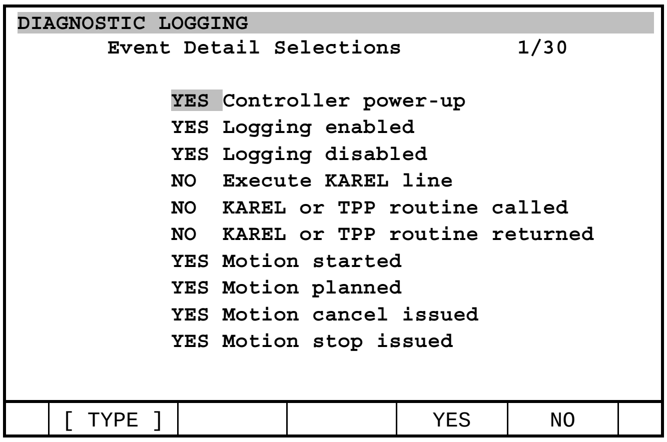
Steps

1 Press [MENU] key.
2 Select TEST CYCLE.
3 Press F1, [TYPE].
4 Select Debug Ctl. The following screen will be displayed.



```
DIAGNOSTIC LOGGING
        Main Menu                        1/9

           Clear event log
           Dump log data
           Add a task to log
           Stop logging a task
           List selected tasks
           Set classes to log
           Change events to log
           Enable all logging
           Disable all logging

   [ TYPE ]                          HELP
```

5 Select Change events to log. The following screen will be displayed.



```
DIAGNOSTIC LOGGING
       Event Detail Selections         1/30

            YES Controller power-up
            YES Logging enabled
            YES Logging disabled
            NO  Execute KAREL line
            NO  KAREL or TPP routine called
            NO  KAREL or TPP routine returned
            YES Motion started
            YES Motion planned
            YES Motion cancel issued
            YES Motion stop issued

   [ TYPE ]                    YES     NO
```

6 Enable or disable a specific event type to be logged.
- To turn on logging for an event type, press F4, YES.
- To turn off logging for an event type, press F5, NO.

The changes take effect immediately.

26.2.8 Enable or Disable All Event Logging

The Enable/Disable All Logging Screen allows you to start or stop the automatic logging of all selected events. When the Program Diagnostic Option is installed, logging of selected events in selected tasks is enabled. The Disable all logging menu selection turns off all logging. It also eliminates the overhead added when logging is enabled. (Refer to Subsection 26.2.4 "Performance".) The Enable all logging menu selection can be used to start logging again.

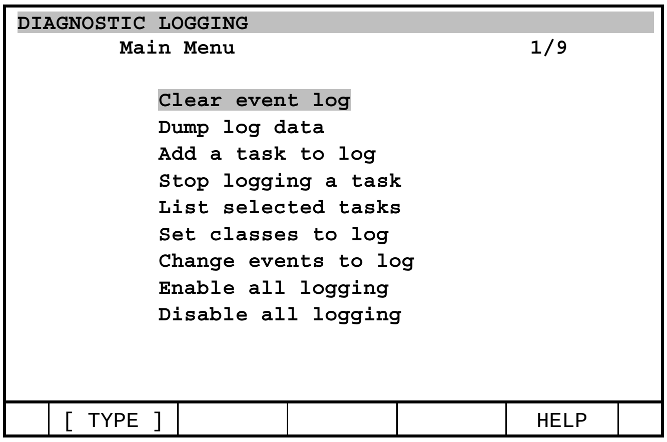
NOTE
Event and task selections are not changed by Disable all logging or Enable all logging.

Use Procedure 26-7 to enable and disable all event logging.

Procedure 26-7 Enabling and Disabling All Event Logging

Steps

1 Press [MENU] key.
2 Select TEST CYCLE.
3 Press F1, [TYPE].
4 Select Debug Ctl. The following screen will be displayed.



```
DIAGNOSTIC LOGGING
        Main Menu                      1/9

           Clear event log
           Dump log data
           Add a task to log
           Stop logging a task
           List selected tasks
           Set classes to log
           Change events to log
           Enable all logging
           Disable all logging
```

5 Select either Enable or Disable all logging.

⚠CAUTION
If you Enable or Disable all logging, you must perform a Cold Start for the change to take effect.

- If you select Enable all logging, selected events in selected tasks will automatically start being logged after you perform a Cold Start.
- If you select Disable all logging, selected events in selected tasks will automatically stop being logged after you perform a Cold Start.

6 Perform a Cold Start.

26.3 LOGGING EVENTS

Events are conditions or situations that occur in a KAREL or teach pendant program while the program is running. The KAREL Program Execution History Record Option allows you to record certain events, as they take place, to help you in debugging programs. The logged events can be dumped to an ASCII file.

26.3.1 Setting up Events

An event will be recorded in the log when all of the following are true:

- You have selected Enable all logging from the main menu.
- You have enabled event logging for the task.
- You have selected the event type to log either,
  - Specifically, using the Change Events To Log screen to select a specific event.

- Generally, using the Set Classes To Log screen to select a class of events.

Types of Events that can be Logged

With the Program Diagnostic Option, the following kinds of events can be recorded. Refer to Section 26.4.4 for additional information.

- Controller power-up
- Logging enabled
- Logging disabled
- Execute KAREL line
- KAREL or TPP routine called
- KAREL or TPP routine returned
- Motion started
- Motion planned
- Motion cancel issued
- Motion stop issued
- Resume move
- Motion done received
- Motion completed normally
- Condition handler triggered
- Condition handler enabled
- Condition handler disabled
- Before processing ISR
- After processing ISR
- Return from KAREL ISR
- KAREL or TPP task starts execution
- KAREL or TPP task aborts
- Interpreter receives packet
- Starting execution of p-code
- Start of execution of TPP line
- MMR received
- Normal AMR received by AMGR
- Start AMR received by AMGR
- Stop AMR received by AMGR
- AMR sent to AX
- AMR received from AX

26.3.2 Logging Events to an ASCII File

The last 500 events that are recorded can be written to an ASCII file. This can be done by requesting to write the logged data to a file on the Dump Selections screen (refer to Subsection 26.2.2).

26.3.3 ASCII File General Event Information

Specific information about each event is logged to the ASCII file. Table 26.3.4 describes the following information for each event:

- **Event name:** Event description displayed in the Event Detail Selections screen.
- **When recorded:** Conditions under which the event is recorded. For example, the STRT_K_LINE event is recorded when a KAREL statement is about to be executed.
- **Information recorded:** List of values logged with each event. The following are reported for all events (except as noted).
  - Event time (seconds) since power-up. This is always a multiple of 4 milliseconds (.004 second).
  - Number of the task associated with the event.
  - Name of the KAREL routine or TPP program, in which the statement was triggered.
  - Program line number at which statement event was triggered.

- **Enabled default:** YES if logging of the specified event type is Enabled by default.
- **Comments:** Additional information, if appropriate.

26.3.4 ASCII File Specific Event Information

Table 26.3.4 contains each event and the corresponding information that is logged.

Table 26.3.4 Event logging information

Event Name	When Recorded	Information Recorded	Enabled Default	Comments
Controller power-up	At every controller COLD start	Routine name, line number Clock time: at which power-up occurred, in "DD-MMM-YY HH:MM" format. Task number will always be 1.	YES	
Logging enabled	When logging is started. Typically, this is when the task starts.	Standard information only	YES	
Logging disabled	When logging is stopped. Typically, this is when the task ends.	Standard information only	YES	
Execute KAREL line	At beginning of execution of KAREL statement.	Internal information Internal information Internal information	NO	
KAREL or TPP routine called	When a KAREL routine or KAREL or TPP program is called. This is either directly or as a result of a condition handler action.	Standard information only	NO	Routine name is name of called routine; line number is line number from which call was made or interrupt occurred.
KAREL or TPP routine returned	When a KAREL PROGRAM or ROUTINE or TP program returns to the calling or interrupted program.	Standard information only	NO	The ROUTINE name is the name of the returning program or routine; the line number is the line in the calling or interrupted program to which the task is returning, generally the same as the line number shown for in the CALL event.
Motion started	When a request to initiate a motion is issued.	Group mask MMR address, in hexadecimal MMR status, in hexadecimal; normally FFFFFFFF	NO	Intended for system level analysis.
Motion planned	For TPP motion commands, this occurs typically before the motion is actually started.	Group mask MMR address, in hexadecimal MMR status, in hexadecimal; normally FFFFFFFF	NO	Intended for system level analysis.

Event Name	When Recorded	Information Recorded	Enabled Default	Comments
Motion cancel issued	When a CANCEL statement or condition handler action is executed, a MOVE... UNTIL condition is satisfied, a CANCEL severity error is posted, or a program is aborted with a motion planned or in progress.	Group mask	NO	Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		
Motion stop issued	When a STOP statement or condition handler action is executed or a STOP severity error is posted.	Group mask	NO	Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		
Resume move	When a stopped motion is resumed; typically when a RESUME statement or condition handler action is executed; also when a program is CONTINUED following a STOP error condition.	Group mask	NO	Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		
Motion done received	When the termination type is satisfied for a KAREL or TPP motion statement is executed. For KAREL MOVE ... NOWAIT statements, this event if recorded when the motion starts.	Group mask	NO	Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		
Motion completed normally	When a motion is completed or cancelled. This is generally after MTN_DONE is recorded.	Group mask	NO	Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		
Condition handler triggered	When a global condition handler triggers.	Condition handler number	NO	Intended for system level analysis.
Condition handler enabled	When a global condition handler is enabled by an ENABLE statement or condition handler action.	Condition handler number	NO	Intended for system level analysis.
Condition handler disabled	When a global condition handler is disabled by a DISABLE statement or condition handler action. It is not recorded when a condition handler triggers.	Condition handler number	NO	Intended for system level analysis.

Event Name	When Recorded	Information Recorded	Enabled Default	Comments
Before processing ISR	Before calling a KAREL PROGRAM from a condition handler.	Internal information	NO	The routine name and line number indicate the code that was executing when the interrupt routine request was received.
		Internal information		
		Internal information		
After processing ISR	When the interrupt routine is ready to run.	Standard information only	NO	Routine name is the name of the interrupt routine. Line number will always be 1.
Return from KAREL ISR	When exiting from an interrupt routine.	Standard information only	NO	Routine name and line number are the interrupt routine name and the line number from which it returned.
KAREL or TPP task starts execution	When a task selected for logging starts executing.	Standard information only	YES	Routine name is the main program name; line number is always 1.
KAREL or TPP task aborts	When a task ends.	Standard information only	YES	Routine and line number indicate the last statement executed by the task. This might be an END statement of a KAREL program or the last line of a TPP program.
Interpreter receives packet	When an interpreter task receives a packet.	Packet address	NO	Intended for system level analysis.
		Packet status		
		Request code (including sub-system code)		
		Requestor id		
		ITR-level		
Starting execution of p-code	At start of execution of KAREL p-code instruction.	Number of words on the routine stack	NO	Intended for system level analysis.
		Number of words on the data stack		
		P-code mnemonic		
Start of execution of TPP line	At start of execution of a line of a TPP program.	Number of words on the data stack	NO	
		Number of words on the routine stack		
		Number of words available for combined ROUTINE and DATA stacks		

Event Name	When Recorded	Information Recorded	Enabled Default	Comments
MMR received	When an MMR is received back from the motion sub-system. This occurs when the motion completes, is cancelled, or is stopped.	Group mask	NO	Intended for system level analysis.
		MMR address, in hexadecimal		
		MMR status, in hexadecimal; normally FFFFFFFF		
Normal AMR recvd by AMGR	An AMR is received by AMGR.	Address of AMR	NO	Intended for system level analysis.
		AMR number		
		AMR AMGR_wk		
		AMR ax_phase		
Start AMR recvd by AMGR	A start AMR request is processed by AMGR.	Address of AMR	NO	Intended for system level analysis.
		AMR number		
		AMR AMGR_wk		
		AMR ax_phase		
Stop AMR recvd by AMGR	A stop AMR request is processed by AMGR.	Address of AMR	NO	Intended for system level analysis.
		AMR number		
		AMR AMGR_wk		
		AMR ax_phase		
AMR sent to AX	When an AMR is sent from AMGR to an AX task.	Address of AMR	NO	Intended for system level analysis.
		AMR number		
		AMR AMGR_wk		
		AMR ax_phase		
AMR rcvd from AX	When an AMR is receivedack from an AX task.	Address of AMR	NO	Intended for system level analysis.
		AMR number		
		AMR AMGR_wk		
		AMR ax_phase		

26.4 EXAMPLES

26.4.1 Overview

This appendix contains example programs and the resulting log data, that show how the KAREL Program Execution History Record Option operates.

- Subsection 26.4.2 contains a KAREL program (T) which calls a teach pendant program (TPP).
- Subsection 26.4.3 contains the teach pendant program that is called (TTT).
- Subsection 26.4.4 contains the log file that is generated after running the KAREL program (T) with the following Event class selections:
  - YES Execute KAREL line
  - YES Execute TPP line
  - YES Call/return
  - YES Motion
  - YES Condition Handler

- YES Interrupt rtn
- YES Task start/end
- NO Packet_rcd
- NO Pcode exec

26.4.2 KAREL Program Example

The KAREL program in Example 26.4.2 is the MAIN program from which the TPP program is called and executed.

Example 26.4.2 KAREL program example (T.KL)

```
program t
var
  i,j,k: INTEGER

routine ttt from ttt
routine tt
begin
  i = i + 1
end tt

begin

condition[1]:
  when k >= 500 DO
    k = 0
    tt
    enable condition[1]
  endcondition
k = 0
i = 0
connect timer to k
enable condition[1]
wait for i=2
disconnect timer k
disable condition[1]
ttt
DELAY 1000
end t
```

26.4.3 Teach Pendant Program Example

The TPP program in Example 26.4.3 is called by the KAREL program (T) in Example 26.4.2.

Example 26.4.3 Teach pendant program example (TTT.TP)

```
1:J P[1] 100% FINE
2:  WAIT   0.00(sec)
3:  R[1]=0
4:  LBL[1]
5:  R[1]=R[1]+1
6:  IF R[1] <= 3,JMP LBL[1]
7:  WAIT    .50(sec)
```

26.4.4 ASCII File Example

With logging task T enable and event classes selected as indicated in Example 26.4.2, if T is executed and the log dumped, the file show in Example 26.4.4 is generated.

NOTE
For detailed information about the fields on the ASCII report shown in Example 26.5.4, refer to Subsection 26.4.4.

Example 26.4.4 ASCII File Example

```
   Event           Time   TID         Routine  Line
 POWER-UP           .104   1              ***   ***   09- 9-28 11:53
 TASK-START       84.888   8                T     1
 LOG-ENABLE       84.888   8                T     1
 STRT-K-LINE      84.888   8                T    13   0    0   19200
 STRT-K-LINE      84.904   8                T    19   0    0   19200
 STRT-K-LINE      84.904   8                T    20   0    0   19200
 STRT-K-LINE      84.904   8                T    21   0    0   19200
 STRT-K-LINE      84.904   8                T    22   0    0   19200
 STRT-K-LINE      84.904   8                T    23   0    0   19200
 CH-ENABLE        84.904   8                T    23   1
 CH-TRIGGER       85.912   8                T    23   1
 PRE-ISR          85.912   8                T    23   0  128      35
 CALL             85.912   8               TT    23
 IN-ISR           85.912   8               TT     1
 STRT-K-LINE      85.912   8               TT     8   0   68    1792
 STRT-K-LINE      85.928   8               TT     9   0   68    1792
 RETURN           85.928   8               TT    23
 RTN-INT-RTN      85.928   8               TT    23
 CH-TRIGGER       86.920   8                T    23   1
 PRE-ISR          86.920   8                T    23   0  128      35
 CALL             86.920   8               TT    23
 IN-ISR           86.920   8               TT     1
 STRT-K-LINE      86.920   8               TT     8   0   68    1792
 STRT-K-LINE      86.936   8               TT     9   0    8    1792
 RETURN           86.952   8               TT    23
 RTN-INT-RTN      86.952   8               TT    23
 STRT-K-LINE      86.952   8                T    24   0    0   19200
 STRT-K-LINE      86.952   8                T    25   0    0   19200
 STRT-K-LINE      86.952   8                T    26   0    0   19200
 CH-DISABLE       86.952   8                T    26   1
 CALL             86.952   8              TTT    26
 STRT-T-LINE      86.952   8              TTT     1   0   16   15104
 PLAN-MOVE        86.952   8              TTT     1   405E61A8   FFFFFFFF   6
 START-MOVE       86.952   8              TTT     1   405E61A8   FFFFFFFF   6
 MTN-DONE         88.088   8              TTT     1   405E61A8   FFFFFFFF   3
 STRT-T-LINE      88.088   8              TTT     2   0   16   15104
 MMR_RCVD         88.104   8              TTT     1   405E61A8   00000000   3
 MTN_ENDED        88.104   8              TTT     1   405E61A8   00000000   3
 STRT-T-LINE      88.104   8              TTT     3   0   16   15104
 STRT-T-LINE      88.104   8              TTT     4   0   16   15104
 STRT-T-LINE      88.104   8              TTT     5   0   16   15104
 STRT-T-LINE      88.104   8              TTT     6   0   16   15104
 STRT-T-LINE      88.136   8              TTT     4   0   16   15104
 STRT-T-LINE      88.136   8              TTT     5   0   16   15104
 STRT-T-LINE      88.136   8              TTT     6   0   16   15104
 STRT-T-LINE      88.168   8              TTT     4   0   16   15104
 STRT-T-LINE      88.168   8              TTT     5   0   16   15104
 STRT-T-LINE      88.168   8              TTT     6   0   16   15104
 STRT-T-LINE      88.200   8              TTT     4   0   16   15104
 STRT-T-LINE      88.200   8              TTT     5   0   16   15104
 STRT-T-LINE      88.200   8              TTT     6   0   16   15104
 STRT-T-LINE      88.200   8              TTT     7   0   16   15104
 RETURN           89.208   8              TTT    26
 STRT-K-LINE      89.208   8                T    27   0    0   19200
 STRT-K-LINE      91.224   8                T    28   0    0   19200
 TASK-END         91.240   8                T    28
 LOG-DISABLE      91.240   8                T    28
```

27 TORQUE LIMIT FUNCTION

Using a TP instruction, Torque Limit function can limit the maximum motor torque as desired. And Torque limit multi-axis setup function can set up torque limit for multi axes respectively. To use these functions, Torque Limit function option (A05B-2600-J611) is required.

You can realize followings by Torque Limit function.
This manual describes Torque Limit function using a Servo Hand which has a servo driven gripper.

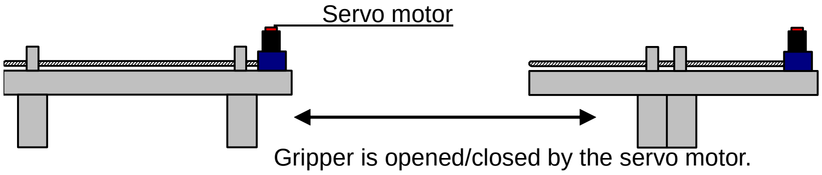


Fig. 27 (a) Example of servo hand

(1) Easy to teach the gripper position

Servo Hand can control the Open/Close position of gripper, but it is necessary to teach gripper position according to the work piece length (size). Then it is necessary to teach position for each work piece type, and it is not easy to operate and maintain the Servo Hand.

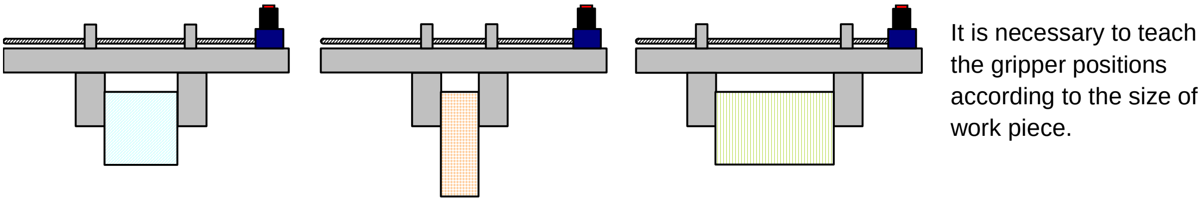

It is necessary to teach the gripper positions according to the size of work piece.

Fig. 27 (b) Teaching by legacy style

This function can stop the gripper movement when the external force (load) of the servo motor exceed the specified motor torque. Therefore if you teach a closed gripper position and use Torque Limit for this motion instruction, the gripper will stop automatically at the position fit to the work length.

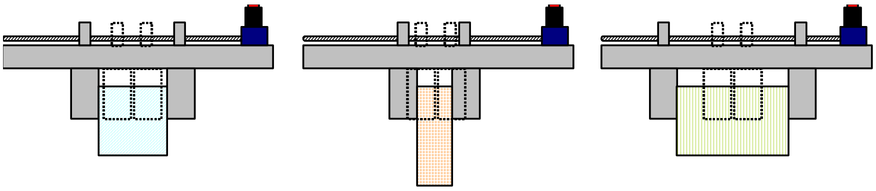


Fig. 27 (c) Teaching by torque limit function

Merit1: Easy to teach

One teaching point (where gripper is closed) can be used for all work pieces.

(2) Gripping force is adjustable.

By TORQ_LIMIT instruction, it is possible to change the maximum motor torque to arbitrary value. Therefore, it is possible to adjust the gripping force to the arbitrary value.

Merit2: Gripping force is adjustable.

It is possible to adjust the gripping force to the arbitrary value because the maximum motor torque can be limited as desired.

27.1 TORQUE LIMIT FUNCTION FEATURE

Torque Limit function can limit the maximum torque of servo motor which is controlled by a robot controller. You can set Torque Limit value in % of the maximum motor torque. Depending on the combination of a motor and an amplifier, the maximum current of the amplifier may be limited to guard the motor. But, in that case, the torque corresponding to the maximum current in the specification of the amplifier is the reference value. For example, in case that the maximum current of the amplifier is limited to 20Ap though the maximum current in the specification is 40Ap, if you want to limit the torque to the torque corresponding to 10Ap, specify 25% to the torque limit because 10Ap is 25% of 40Ap.

In case that the torque limit value which exceeds the limit for motor guard is specified by the torque limit instruction, the maximum current is limited automatically when executing the torque limit instruction so that actually the maximum current does not exceed the limit for motor guard. Then "INTP-271 Excessive torque limit value" warning is also shown. In above example, even if the value greater than 50% is specified to the torque limit, the maximum current is clamped to 50% which is correspond to 20Ap automatically.

Syntax ... TORQ_LIMIT t%   (t: 0.0 to 100.0)

Using following system variables, you can select the axes to be limited the motor torque. (By default, J7 axis in Group 1 is enabled.)

-Group number

$TORQUE_LMT.$GROUP[i]   Select the group number including the servo motor using Torque Limit.
i : Group number

-Axis number

$TORQUE_LMT.$GAi[j]   Select the group number and axis number of the servo motor using Torque Limit. i : Group number, j : Axis number

For example, if J1 axis in Group 2 uses Torque Limit, set TRUE to $G2[1], and set others FALSE.

```
$TORQUE_LMT.$GROUP[2]  = TRUE
$TORQUE_LMT.$GA2[1]    = TRUE
$TORQUE_LMT.$GA2[2]    = FALSE
:
$TORQUE_LMT.$GA2[9]    = FALSE
```

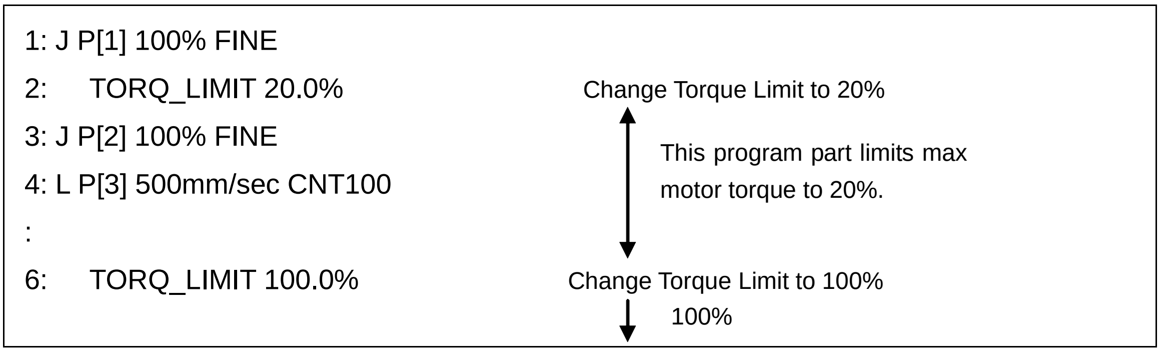
You can set multiple axes to TRUE.

In the case of the dual drive, set TRUE to both the master axis and slave axis.

⚠ **CAUTION**
Usually GA1[1] - [6] are robot axes. So, please do NOT set these variables to TRUE.

⚠ **CAUTION**
Check the intended axes of Torque Limit function in all axes in all groups that exist in your robot controller.

It is necessary for you to decide Torque Limit value.
It is necessary to decide Torque Limit value by gripping actual work piece.
After a TORQ_LIMIT instruction is executed, the maximum motor torque will not be changed until another TORQ_LIMIT instruction will be executed.



```
1: J P[1] 100% FINE
2:    TORQ_LIMIT 20.0%
3: J P[2] 100% FINE
4: L P[3] 500mm/sec CNT100
:
6:    TORQ_LIMIT 100.0%
```

Change Torque Limit to 20%
This program part limits max motor torque to 20%.
Change Torque Limit to 100%
100%

Fig.27.1 Example program of torque limit function

After decreasing maximum motor torque by TORQ_LIMIT instruction, the gripper axis may not reach the destination point.

It is necessary to set following system variables to large value in advance.
$PARAM_GROUP[group].$STOPTOL[axis]
$PARAM_GROUP[group].$STOPERLIM[axis]
$PARAM_GROUP[group].$MOVER_OFFST[axis]

Setting the upper limit of Torque Limit
The upper limit of Torque Limit can be specified by the following system variables.
$TORQUE_LMT.$MAX_TRQ_LMT = Upper limit (0.1 - 100.0)
The lower limit is fixed to 0.1.

27.2 TORQUE LIMIT MULTI-AXIS SETUP FUNCTION

27.2.1 Torque Limit Multi-Axis Setup Function

Please set system variable $karel_enb to TRUE in order to select karel at "Call" instruction.
You can set torque limit to multi-axis using "CALL" from TP program.
Following explains how to use from TP program.

```
CALL TPTRQLIM(group number, axis number, torque limit value(%))
```

Fig. 27.2.1 (a) How to call “Torque limit multi-axis setup function”

- group number : Set the group number including the servo motor using Torque Limit.
- axis number : Set the axis number of the servo motor using Torque Limit.
- torque limit value : Set Torque Limit value in % of the maximum motor torque.
(You can set 0.1% to 100.0%. You can also use integer.)

> ⚠ **CAUTION**
> Do not set robot axis to axis number.

Following is an example TP problem.(This example TP program set 70% torque limit to axis 7 of group 1.)

```
CALL TPTRQLIM(1, 7, 70)
```

Fig. 27.2.1 (b) Example program of Torque limit multi-axis setup function

In the case of the dual drive, set TRUE to both the master axis and slave axis.

27.3 LIMITATIONS

- This function limits the maximum motor torque of the specified servo motor, but this function does NOT guarantee the motor torque while rotating the servo motor.
- Because this function is designed for FANUC servo motor, if you want to use this function to non FANUC servo motor, contact your local FANUC representative.

27.4 CAUTION

- It is possible to use Torque Limit for robot axes, but please do NOT enable this function for any robot axes which may be fallen down by the gravity. (Torque Limit multi-axis setup function is not applied to the motor of the robot axes.)
- When an axis is stopped by Torque Limit, the external force is equal to the limited motor torque, and the distance between current position and destination position is considered as position error. (Please see the highlighted part in the figure below.) If you change Torque Limit to 100%, the axis moves to the destination with 100% of motor torque, therefore you may break Servo Hand or a work piece. If you want to change Torque Limit to 100% (or increase motor torque), you have to move the axis where above problem will not occur before changing Torque Limit.

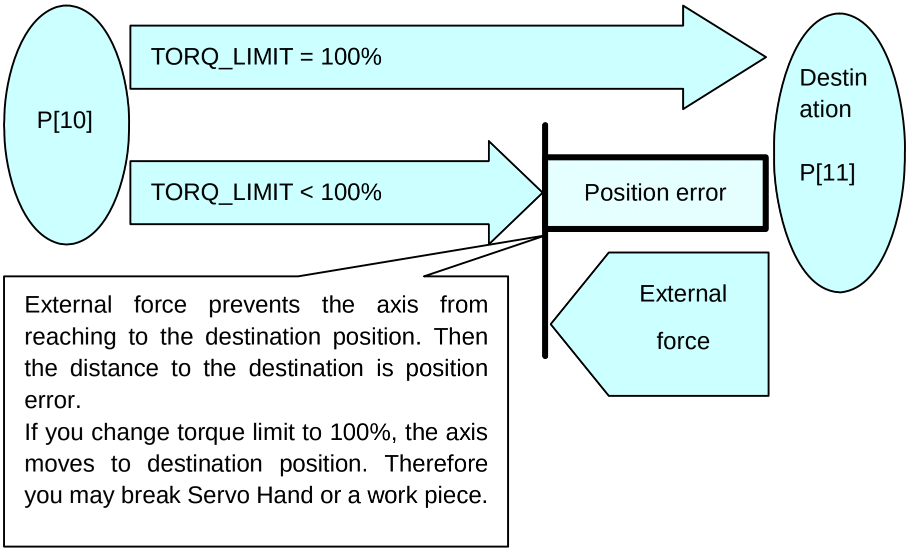

P[10]
TORQ_LIMIT = 100%
TORQ_LIMIT < 100%
Position error
Destin ation
P[11]
External force
External force prevents the axis from reaching to the destination position. Then the distance to the destination is position error.
If you change torque limit to 100%, the axis moves to destination position. Therefore you may break Servo Hand or a work piece.

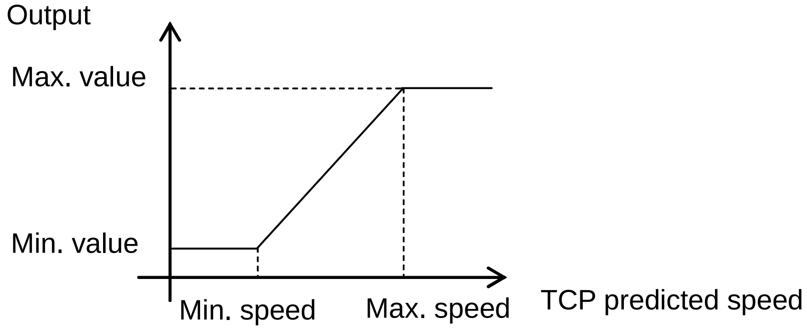
Fig. 27.4 Caution for torque limit function

28 TCP SPEED OUTPUT

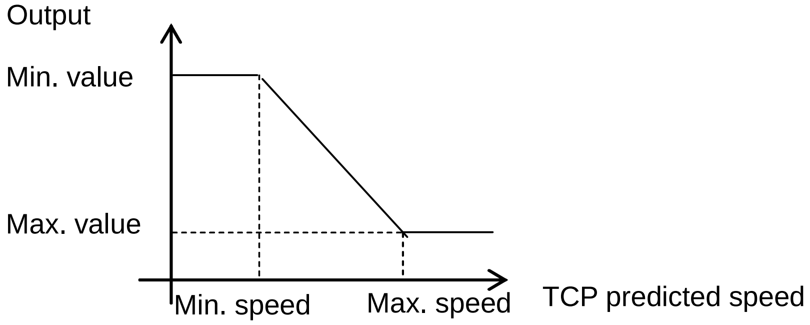
The TCP speed output function outputs the predicted speed of tool center point (TCP) via analog output signals, group output signals, or registers. A software option, TCP speed output function, is required to use this function. The TCP predicted speed is converted on the output condition that is defined by the following setup items:

- Min. value
- Max. value
- Min. speed (mm/sec)
- Max. speed (mm/sec)

The following figure shows the relationship between TCP predicted speed and output.



The Min value is output when TCP predicted speed is the specified Min speed or below. The Max value is output when TCP predicted speed is the specified Max speed or above. The following figure shows the relationship between TCP predicted speed and output when you set the Max value to a value of less than Min value.



28.1 LIMITATIONS

There are limitations on use of the TCP speed output function.

- This function is not available on DispenseTool or Dispense-enabled SpotTool+.
- This function is not available with tracking functions e.g. Line Tracking.
- TCP predicted speed is output only while the robot is moving by program execution.
- TCP predicted speed reflects only group 1 motion.
- TCP predicted speed does not reflect the extended axis motion except the integrated axis one.
- Accuracy of TCP predicted speed may lower when the motor speed limit alarms are posted.
- Accuracy of TCP predicted speed may lower while the frame is being switched.
- Accuracy of TCP predicted speed may lower while the motion is being switched between the normal and the remote TCP motion.
- Accuracy of the TCP predicted speed may lower while one of the following instructions is being executed:

WAIT instruction
Joint motion instruction
Motion instruction with Approach/Retract (Linear Distance) instruction
Motion instruction with Process Speed instruction
Motion instruction with Max Speed instruction
Motion control statement in KAREL program

- Up to 10 output conditions can be defined.
- Only one prediction time can be specified. Different prediction times cannot be specified among output conditions.

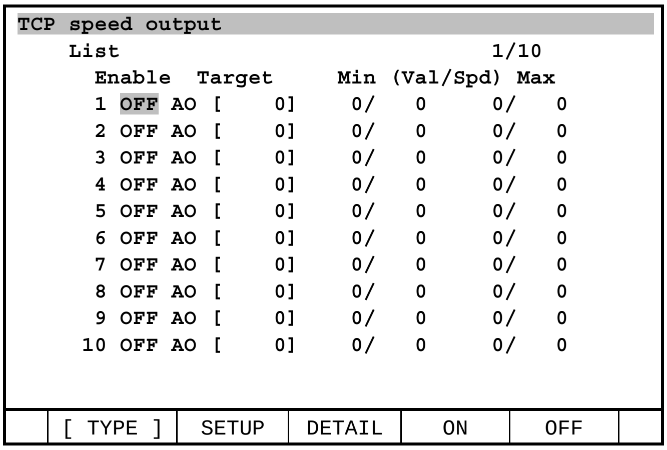
28.2 SETTING UP TCP SPEED OUTPUT

To use the TCP speed output function, you must set up the output condition using the procedure 28-1.

Procedure 28-1 Setting up TCP speed output conditions

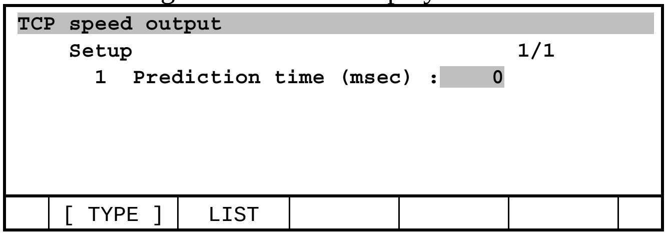
Step

1 Press [MENU] key.
2 Select I/O.
3 Press F1, [TYPE].
4 Select TCP Speed. The following screen will be displayed.



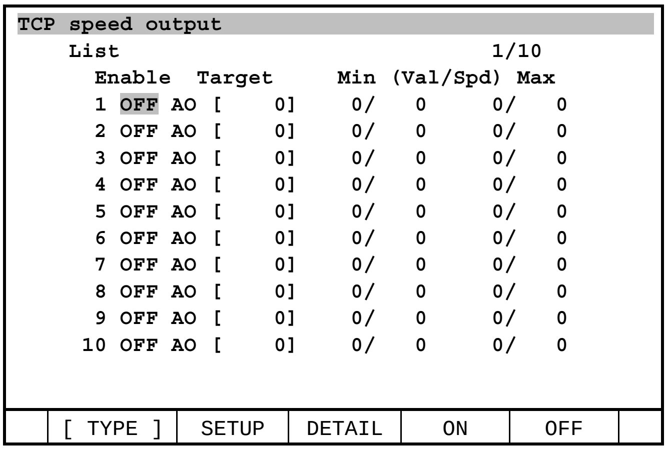
```
TCP speed output
    List                              1/10
      Enable  Target      Min (Val/Spd) Max
      1 OFF AO [     0]     0/   0      0/   0
      2 OFF AO [     0]     0/   0      0/   0
      3 OFF AO [     0]     0/   0      0/   0
      4 OFF AO [     0]     0/   0      0/   0
      5 OFF AO [     0]     0/   0      0/   0
      6 OFF AO [     0]     0/   0      0/   0
      7 OFF AO [     0]     0/   0      0/   0
      8 OFF AO [     0]     0/   0      0/   0
      9 OFF AO [     0]     0/   0      0/   0
     10 OFF AO [     0]     0/   0      0/   0
```

5 Press F2, SETUP. The following screen will be displayed.



```
TCP speed output
    Setup                               1/1
      1  Prediction time (msec) :     0
```

6 Set the value for Prediction time.
7 Press F2, LIST. The following screen will be displayed.

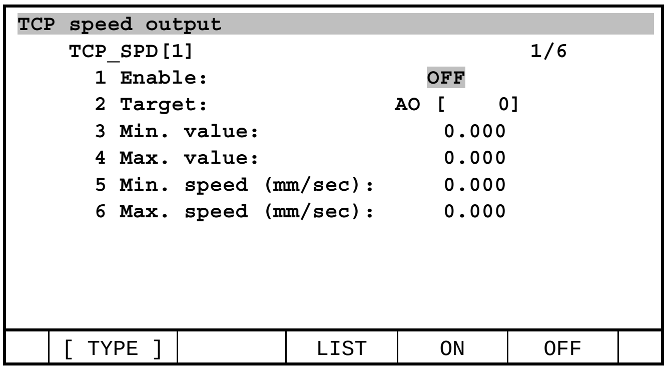


```
TCP speed output
    List                                1/10
      Enable  Target     Min (Val/Spd) Max
      1 OFF AO [     0]     0/   0      0/   0
      2 OFF AO [     0]     0/   0      0/   0
      3 OFF AO [     0]     0/   0      0/   0
      4 OFF AO [     0]     0/   0      0/   0
      5 OFF AO [     0]     0/   0      0/   0
      6 OFF AO [     0]     0/   0      0/   0
      7 OFF AO [     0]     0/   0      0/   0
      8 OFF AO [     0]     0/   0      0/   0
      9 OFF AO [     0]     0/   0      0/   0
     10 OFF AO [     0]     0/   0      0/   0
```

NOTE

Each line in this screen indicates the status of one output condition.

8 Move the cursor to an output condition that you want to set up, and press F3, DETAIL. The following screen will be displayed.



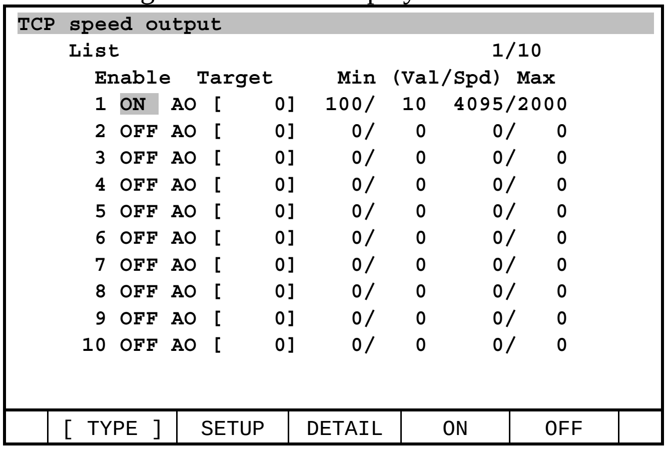
```
TCP speed output
    TCP_SPD[1]                               1/6
      1 Enable:                      OFF
      2 Target:                   AO [    0]
      3 Min. value:                   0.000
      4 Max. value:                   0.000
      5 Min. speed (mm/sec):          0.000
      6 Max. speed (mm/sec):          0.000
```

9 Move the cursor to Target and select the output target.
10 Move the cursor to each item and set the value.
11 To start the output, set Enable to ON.

NOTE

You can also set Enable and Target by using an instruction. Refer to Procedure 28-2 for more information.

12 Press F3, LIST. The following screen will be displayed.



```
TCP speed output
    List                                1/10
      Enable  Target     Min (Val/Spd) Max
      1 ON  AO [     0]   100/  10   4095/2000
      2 OFF AO [     0]     0/   0      0/   0
      3 OFF AO [     0]     0/   0      0/   0
      4 OFF AO [     0]     0/   0      0/   0
      5 OFF AO [     0]     0/   0      0/   0
      6 OFF AO [     0]     0/   0      0/   0
      7 OFF AO [     0]     0/   0      0/   0
      8 OFF AO [     0]     0/   0      0/   0
      9 OFF AO [     0]     0/   0      0/   0
     10 OFF AO [     0]     0/   0      0/   0
```

NOTE

You can set Enable for each output condition on this screen.

28.3 TCP SPEED OUTPUT INSTRUCTION

You can set the Enable and Target of TCP speed output function by using an instruction.
The following instruction is used to start the output.

Target type[number]=(TCP_SPD[output condition number])

The following instruction is used to terminate the output.

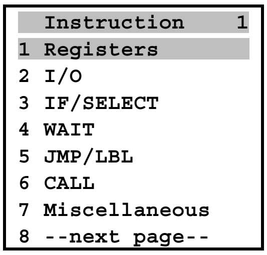
Target type[number]=constant value

Procedure 28-2 directs how to insert the TCP speed output instruction for a register type target.

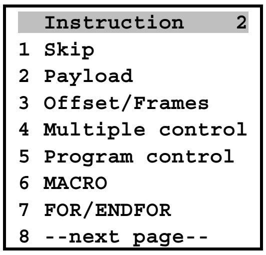
Procedure 28-2 Inserting TCP speed output instruction

Step

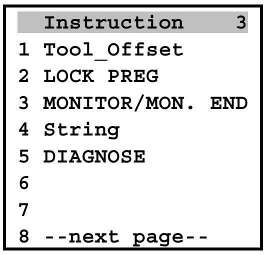
1 Open the program edit screen.
2 Press [NEXT] key.
3 Press F1, [INST]. The following menus will be displayed.



```
  Instruction    1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Miscellaneous
8 --next page--
```

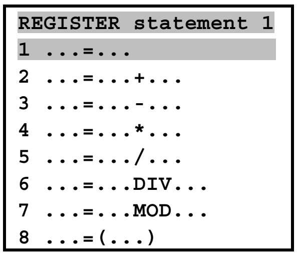


```
  Instruction    2
1 Skip
2 Payload
3 Offset/Frames
4 Multiple control
5 Program control
6 MACRO
7 FOR/ENDFOR
8 --next page--
```



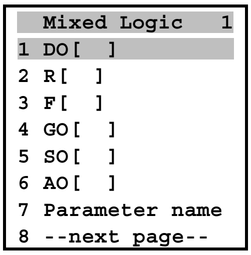
```
  Instruction    3
1 Tool_Offset
2 LOCK PREG
3 MONITOR/MON. END
4 String
5 DIAGNOSE
6
7
8 --next page--
```

4 Select Registers. The following menu will be displayed.

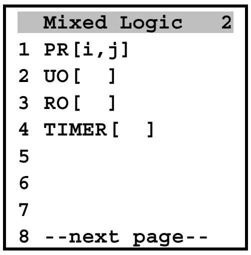


```
REGISTER statement 1
1 ...=...
2 ...=...+...
3 ...=...-...
4 ...=...*...
5 ...=.../...
6 ...=...DIV...
7 ...=...MOD...
8 ...=(...)
```

5 Select …=(...). The following menus will be displayed.

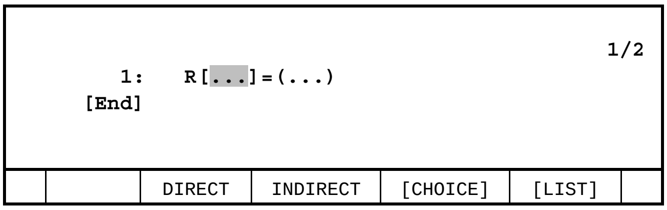


```
  Mixed Logic   1
1 DO[  ]
2 R[  ]
3 F[  ]
4 GO[  ]
5 SO[  ]
6 AO[  ]
7 Parameter name
8 --next page--
```



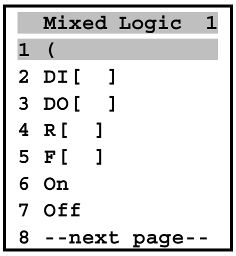
```
  Mixed Logic   2
1 PR[i,j]
2 UO[  ]
3 RO[  ]
4 TIMER[  ]
5
6
7
8 --next page--
```

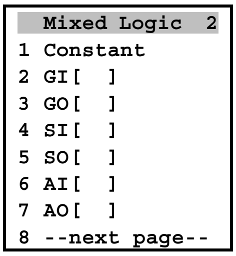
6 Select R[ ].

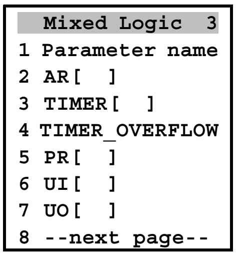


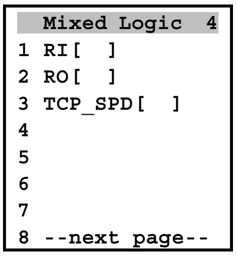
```
                                    1/2
   1:   R[...]=(...)
[End]
```

7 Set the register number for the output. The following menus are displayed.

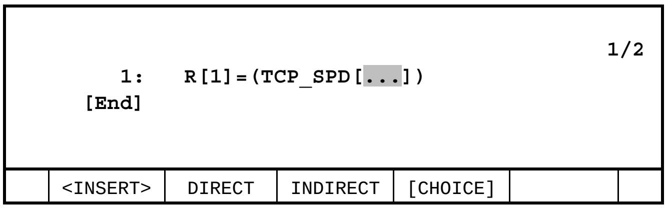






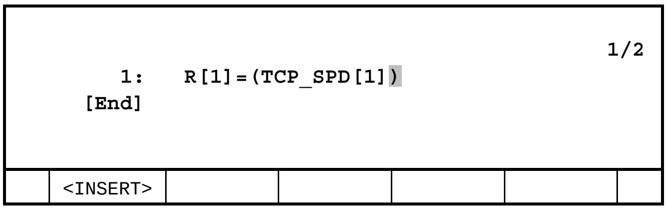


8 Select TCP_SPD[…].



```
                                    1/2
   1:   R[1]=(TCP_SPD[...])
[End]
```

9 Set the output condition number.



```
                                    1/2
   1:   R[1]=(TCP_SPD[1])
[End]
```

29 TP DRAM/FILE STORAGE FUNCTION

TP DRAM/FILE Storage function enhances the capacity of program space by providing for storages outside of CMOS. In 1 motion group system, this function extends the available program space from 15000 points now to 100000 points. However, the maximum points will be fewer than 100000 points depending on system configuration. “R709 TP DRAM/FILE Storage” option is necessary to use this function.

29.1 STORAGES

There are 4 kinds of storages, SHADOW, SHADOW ONDEMAND, FILE, and the existing storage CMOS. The “temporary” storage states mean the transition state when a program is moved between memories for edit or load of a program. The following tables are for the storages and its description.

Table 29.1(a) Storages

Storage	Description
CMOS	Stored into CMOS (non-volatile memory) in the same way as normal controller.
SHADOW	Program is in FROM (non-volatile memory). Loaded into DRAM (volatile memory) on startup.
SHADOW ONDEMAND	Program is in FROM. Loaded into DRAM when accessed.
FILE	Program is in a TP file in the directory specified in “File Path” in Program Configuration screen. Loaded into DRAM when accessed.

Table 29.1(b) Temporary storage status

“temporary” storage state	Description
SHAD (DRAM) ONDE	SHADOW ONDEMAND program is loaded into DRAM.
FILE (DRAM)	FILE program is loaded into DRAM.
SHAD (CMOS)	SHADOW program moved to CMOS for editing.
SHAD (CMOS) ONDE	SHADOW ONDEMAND program moved to CMOS for editing.
FILE (CMOS)	FILE program moved to CMOS for editing.

29.1.1 CMOS Programs

CMOS programs use CMOS (non-volatile memory) as the storage memory.
The programs in other storage are moved to CMOS at editing program temporarily.
The largest program size is dependent on CMOS size.

29.1.2 SHADOW Programs

SHADOW programs use FROM and DRAM as storage memory. As the size of DRAM is larger than that of CMOS, more programs than before will be available by using SHADOW. The capacity for programs that can be used for SHADOW is dependent on the rest capacity of FROM and DRAM.
A program stored in SHADOW is loaded from FROM into DRAM on start up. A program will be executable by loaded into DRAM.
The start up time will be long if there are too many SHADOW programs.
When SHADOW program is edited, it is moved to CMOS and to be “temporary” storage state SHAD (CMOS).

29.1.3 SHADOW ONDEMAND Programs

SHADOW ONDEMAND programs use FROM and DRAM as storage memory. SHADOW ONDEMAND is different form SHADOW in that SHADOW ONDEMAND is loaded when a program is accessed for edit or execution of the program. So a SHADOW ONDEMAND program has no effect to start up time.
The capacity of programs that can be used for SHADOW ONDEMAND is also dependent on the rest capacity of FROM and DRAM as with SHADOW.
When a SHADOW ONDEMAND program is loaded, its storage is moved to “temporary” storage state, SHAD(DRAM) ONDE. There is a slight time lag in the first access, but after that there is no time lag because SHAD(DRAM) ONDE, once loaded into DRAM, remains in DRAM until the power is off.
When a SHADOW ONDEMAND program is edited, its storage is moved to CMOS and to be “temporary” storage state SHAD (CMOS) ONDE.

29.1.4 FILE Programs

FILE programs use storage devices such as memory card as storage memory. In FILE storage, much more programs than in SHADOW can be available by using external storage devices that have large capacity. A FILE program is loaded into DRAM when first accessed as with SHADOW ONDEMAND.
The path to external storage device is specified in “File Path” in Program Configuration screen.
When a FILE program is loaded, its storage is moved into “temporary” storage state, FILE (DRAM).
The time lag in the first access is longer than that of SHADOW ONDEMAND programs, but after that there is no time lag as with SHADOW ONDEMAND because FILE (DRAM) remains in DRAM until the power is off.
When a FILE program is edited, its storage is moved to “temporary” storage state, FILE (CMOS).

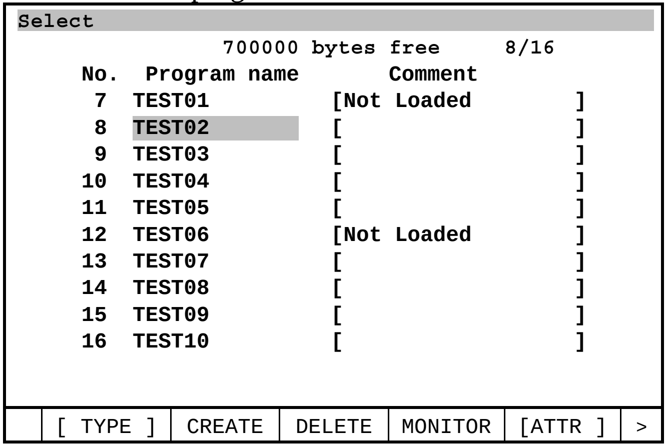
NOTE
Use a dedicated directory in external device for FILE storage path. Do not use shared directly to save TP program files in other storage and backup files.

29.2 STORAGE CONFIGURATION

Procedure 29-1 Confirm storage device

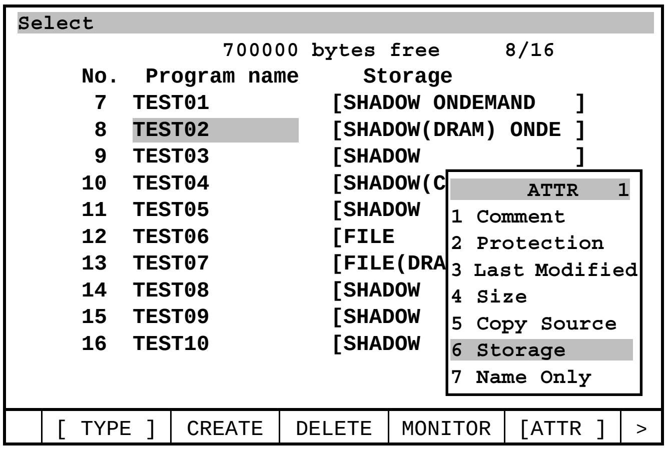
Step

1 Press [SELECT] key. The following screen will be displayed. “Not Loaded” in the comment means that a SHADOW ONDEMAD or FILE program is not loaded into DRAM.



```
Select
                 700000 bytes free     8/16
     No.  Program name       Comment
      7  TEST01          [Not Loaded        ]
      8  TEST02          [                  ]
      9  TEST03          [                  ]
     10  TEST04          [                  ]
     11  TEST05          [                  ]
     12  TEST06          [Not Loaded        ]
     13  TEST07          [                  ]
     14  TEST08          [                  ]
     15  TEST09          [                  ]
     16  TEST10          [                  ]

 | [ TYPE ] | CREATE | DELETE | MONITOR | [ATTR ] | >
```

2 Press F5, [ATTR], and select “Storage”. The following screen will be displayed.



```
Select
                 700000 bytes free     8/16
     No.  Program name     Storage
      7  TEST01          [SHADOW ONDEMAND   ]
      8  TEST02          [SHADOW(DRAM) ONDE ]
      9  TEST03          [SHADOW            ]
     10  TEST04          [SHADOW(C
     11  TEST05          [SHADOW
     12  TEST06          [FILE
     13  TEST07          [FILE(DRA
     14  TEST08          [SHADOW
     15  TEST09          [SHADOW
     16  TEST10          [SHADOW
```

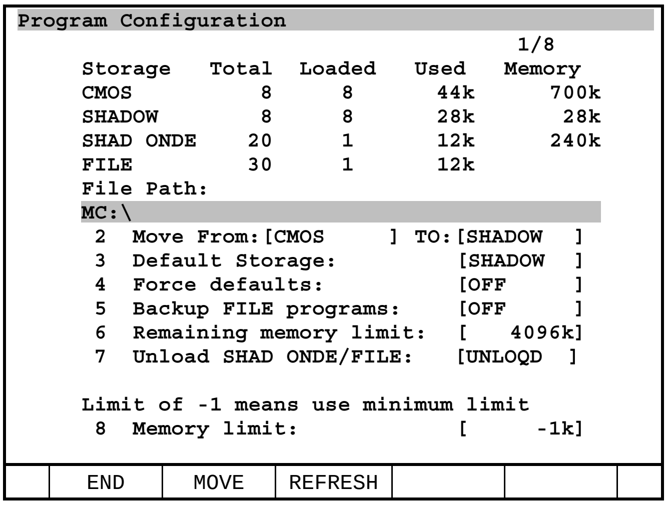
Procedure 29-2 Display Program Configuration screen

Step

1 Press [SELECT] key.

2 Press F1, [TYPE], and select “Config…”.

3 The following screen will be displayed.



```
Program Configuration
                                     1/8
     Storage   Total  Loaded   Used   Memory
     CMOS          8      8      44k      700k
     SHADOW        8      8      28k       28k
     SHAD ONDE    20      1      12k      240k
     FILE         30      1      12k
     File Path:
     MC:\
      2  Move From:[CMOS     ] TO:[SHADOW   ]
      3  Default Storage:         [SHADOW   ]
      4  Force defaults:          [OFF      ]
      5  Backup FILE programs:    [OFF      ]
      6  Remaining memory limit:  [   4096k]
      7  Unload SHAD ONDE/FILE:   [UNLOQD  ]

     Limit of -1 means use minimum limit
      8  Memory limit:            [     -1k]
```

The table below describes the each item in Program Configuration screen.

Table 29.2 Program configuration screen

Item	Description
Storage	For the 4 storages, the number of programs and memory usage are displayed. Total : The number of programs that exist in the storage. Loaded: The number of programs that are actually loaded in the storage. Used : The memory usage for loaded programs. Memory: For CMOS, the capacity for programs is displayed. For others, the total size of the programs that exist in the storage.
File Path	Specify the device where FILE programs are stored.
Move From To	To and from locations for moving programs between storage types.
Default Storage	Set the storage for the new programs. Select from CMOS, SHADOW and SHADOW ONDEMAND. FILE storage cannot be selected.
Force defaults	If valid, loaded into the storage specified in the “Default Storage”. If invalid, loaded into the storage specified in the program file.
Backup FILE programs	Exclude or include FILE program types in backup.

Item	Description
Remaining memory limit	If remaining DRAM goes lower than the value set in “Remaining memory limit” in some operations such as creating a new program, an alarm occurs. You can set the value from 1024 to 10000.The default value is 4096. Usually, please do not change this value.
Unload SHAD ONDE/FILE	Memory of all SHADOW ONDEMAND and FILE programs that loaded to DRAM is released. This function is used to load other programs when DRAM memory is low. Move cursor to this item and push F5 [UNLOAD] to execute this function. These programs are deleted temporarily and they are loaded to DRAM again when they are selected. This function can be used after 7dc2 software.
Memory limit	This item decides the total memory size of SHADOW, SHADOW ONDEMAND and FILE programs that can be loaded into DRAM. If the total memory size goes over the setting value, an alarm occurs. You can set the value from –1 to 10000. The default value is –1. So there is no limitation. Usually, please do not change this value.
END	Finish Program Configuration screen and go back to Select screen.
MOVE	Move programs at a time according to the storage types set in “Move From To”.
REFRESH	Update FILE program list according to TP files in the directory specified in “File Path”. After this operation, there are updated FILE programs displayed in Select screen.

NOTE

The usage of TEMP is different from that displayed in Program Configuration screen. When you load a program into DRAM, it uses three times as much TEMP as the value of the memory displayed in Program Configuration screen.

NOTE

When Default Storage is SHADOW, storage of new programs is set SHADOW. But storage of program could be set CMOS automatically because of shortage of DRAM memory. In this case, please delete unnecessary programs in DRAM. And please recreate the program or change storage individually.

NOTE

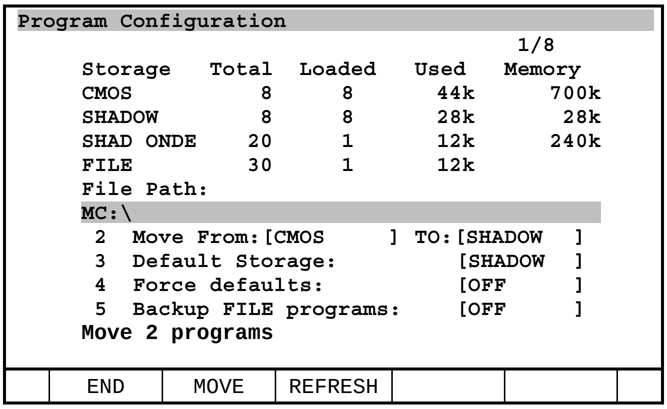
These settings are reflected soon after changed.

Procedure 29-3 Move Storage

Step

- You can move all programs in the same storage to other storage at a time by using “MOVE” in Program Configuration screen.

1. In the item “Move From To”, select the original storage in “From”, and select the new storage in “To”.
2. Press F2, MOVE.
3. Press F4, YES.
4. The following screen will be displayed and moving the storage finished.

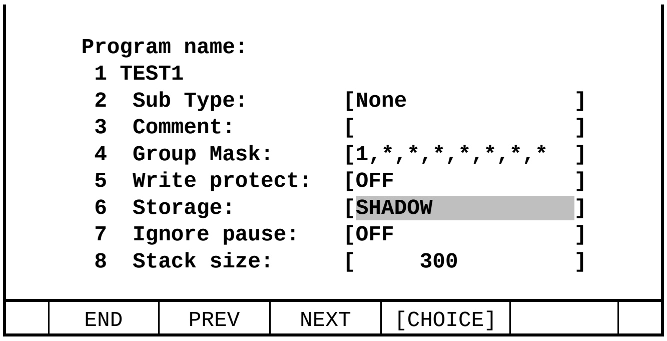


```
Program Configuration
                                        1/8
     Storage    Total  Loaded   Used   Memory
     CMOS           8      8     44k     700k
     SHADOW         8      8     28k      28k
     SHAD ONDE     20      1     12k     240k
     FILE          30      1     12k
     File Path:
     MC:\
      2  Move From:[CMOS     ] TO:[SHADOW   ]
      3  Default Storage:         [SHADOW   ]
      4  Force defaults:          [OFF      ]
      5  Backup FILE programs:    [OFF      ]
     Move 2 programs
```

> **NOTE**
>
> Please confirm that there is enough memory space available in the new storage. If not, an alarm occurs and stops moving. Then, programs that are already moved do not go back the original storage. When you move large number of programs, please make backup in advance.

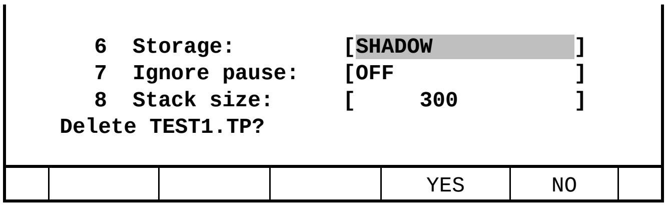
- You can also change the storage of a program individually.

1 Press SELECT key and press NEXT key.
2 Press F2, DETAIL and the following screen will be displayed.
3 Move the cursor on Storage and press [ENTER] key, and select the storage.



```
     Program name:
      1 TEST1
      2  Sub Type:        [None              ]
      3  Comment:         [                  ]
      4  Group Mask:      [1,*,*,*,*,*,*,*   ]
      5  Write protect:   [OFF               ]
      6  Storage:         [SHADOW            ]
      7  Ignore pause:    [OFF               ]
      8  Stack size:      [     300          ]
```

4 When the storage is changed from FILE, “Delete <file name>?” is displayed.
If F4 YES is selected, program file on file path is deleted.
If F5 NO is selected, program file on file path is remained.



```
      6  Storage:         [SHADOW            ]
      7  Ignore pause:    [OFF               ]
      8  Stack size:      [     300          ]
   Delete TEST1.TP?
```

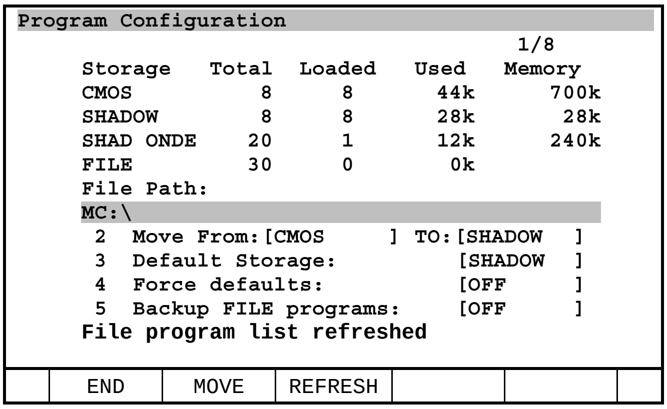
> **NOTE**
>
> Sometimes, the storage goes back to original storage soon after you change the storage. In such case, please confirm if the program is in edit, there is enough memory space in the new storage, or “File Path” is specified if FILE.

Procedure 29-4 Update FILE program list

Step

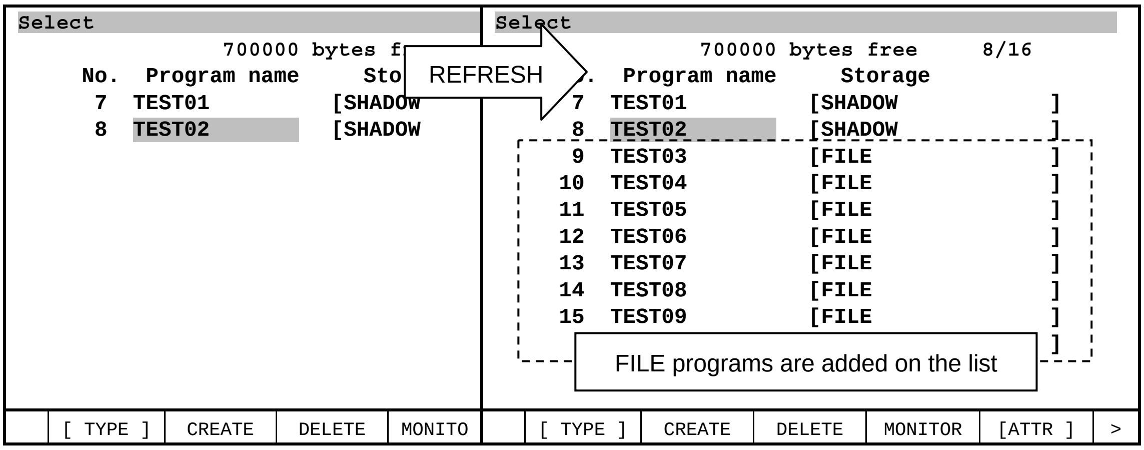
- You can update the FILE program list. This operation is used to display FILE programs again or to use programs in other external storage devices.

1 Input the path of the device where FILE programs are stored, and press ENTER.
2 Press F3, REFRESH.
3 Such screen as follows is displayed, and FILE program list is updated.



```
Program Configuration
                                            1/8
     Storage   Total  Loaded   Used   Memory
     CMOS          8       8    44k     700k
     SHADOW        8       8    28k      28k
     SHAD ONDE    20       1    12k     240k
     FILE         30       0     0k
     File Path:
     MC:\
      2  Move From:[CMOS     ] TO:[SHADOW   ]
      3  Default Storage:         [SHADOW   ]
      4  Force defaults:          [OFF      ]
      5  Backup FILE programs:    [OFF      ]
     File program list refreshed

    END     MOVE     REFRESH
```

Updated FILE programs are listed in Select screen.



```
Select
                700000 bytes f
     No.  Program name      Sto
      7  TEST01          [SHADOW
      8  TEST02          [SHADOW

   [ TYPE ]  CREATE  DELETE  MONITO
```

REFRESH

```
Select
                700000 bytes free     8/16
       .  Program name     Storage
      7  TEST01          [SHADOW              ]
      8  TEST02          [SHADOW              ]
      9  TEST03          [FILE                ]
     10  TEST04          [FILE                ]
     11  TEST05          [FILE                ]
     12  TEST06          [FILE                ]
     13  TEST07          [FILE                ]
     14  TEST08          [FILE                ]
     15  TEST09          [FILE                ]
                                              ]

   [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR ]  >
```

FILE programs are added on the list

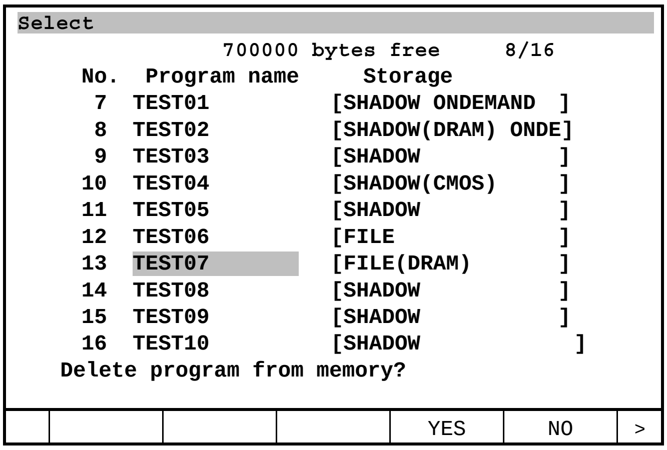
Procedure 29-5 Delete FILE, SHADOW ONDEMAND program

Step

- User programs can be deleted in Select screen. When operation to delete CMOS and SHADOW programs is executed, all program date is deleted.
  For example, when “SHADOW (CMOS)” program is deleted in Select screen, program data in CMOS and DRAM and program in FROM are deleted.
- For FILE and SHADOW ONDEMAND programs, you can choice to delete all program data or only temporary program data, in CMOS and DRAM.

Please refer to the following process to delete FILE and SHADOW ONDEMAND program.

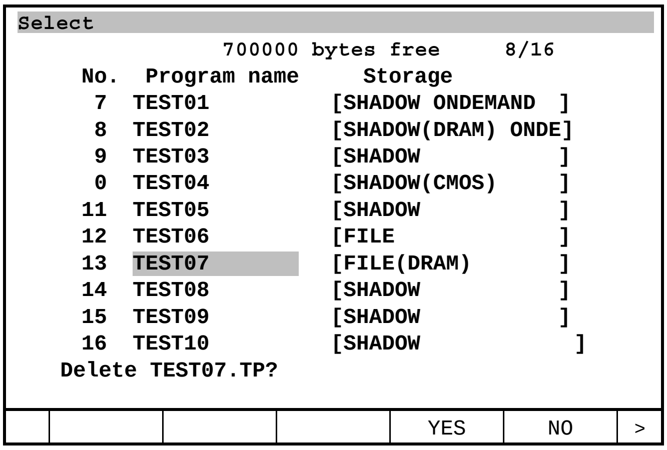
1 Move cursor to the program and press F3 DELETE.
2 When program is loaded to DRAM, “Delete program from memory?” is displayed.



```
Select
                 700000 bytes free     8/16
     No.  Program name     Storage
      7  TEST01          [SHADOW ONDEMAND  ]
      8  TEST02          [SHADOW(DRAM) ONDE]
      9  TEST03          [SHADOW           ]
     10  TEST04          [SHADOW(CMOS)     ]
     11  TEST05          [SHADOW           ]
     12  TEST06          [FILE             ]
     13  TEST07          [FILE(DRAM)       ]
     14  TEST08          [SHADOW           ]
     15  TEST09          [SHADOW           ]
     16  TEST10          [SHADOW             ]
   Delete program from memory?
```

3 If F4 YES is selected, program data in DRAM is deleted.
If F5 NO is selected, operation is terminated and program is not deleted.

4 If F4 YES is selected, following message is displayed continually.
For FILE programs, “Delete <file name>?” is displayed.
For SHDOW ONDEMAND programs, “Delete permanently?” is displayed.



```
Select
                 700000 bytes free     8/16
     No.  Program name     Storage
      7  TEST01          [SHADOW ONDEMAND  ]
      8  TEST02          [SHADOW(DRAM) ONDE]
      9  TEST03          [SHADOW           ]
      0  TEST04          [SHADOW(CMOS)     ]
     11  TEST05          [SHADOW           ]
     12  TEST06          [FILE             ]
     13  TEST07          [FILE(DRAM)       ]
     14  TEST08          [SHADOW           ]
     15  TEST09          [SHADOW           ]
     16  TEST10          [SHADOW             ]
   Delete TEST07.TP?
```

5 If F4 YES is selected, list of the programs in Select screen are also deleted.
For FILE programs, program files on file path are also deleted.
For SHDOW ONDEMAND programs, programs in DRAM are also deleted.
If F5 NO is selected, list of the programs in Select screen are remained. And program files on file path or programs in DRAM are not deleted. When the programs are accessed, they can be loaded into DRAM again because programs on file path or in DRAM are remained.

29.3 SAVE / LOAD PROGRAMS

29.3.1 Save / Load TP Files

When a program is saved as TP file, information for storage is also preserved in it. So, when a TP file is loaded, the program is stored in the saved storage according to the information.
When “Force defaults” is “ON” in Program Configuration screen, the program is stored in the storage specified in “Default Storage”.
When the existing TP file that has no information for storage is loaded, the program is stored in the storage specified in “Default Storage”.

29.3.2 Copy Programs

When a program is copied, the storage of copied program is the same as that of original program.
However, when "Force defaults" is "ON" in Program Configuration screen, copied program is stored in the storage specified in "Default Storage".

29.3.3 Save / Load LS Files

When a program is saved in ASCII format, a LS file is created. Then, new item "STORAGE" is added in the LS file. The storage of the program is printed in this item. When this LS file is loaded, the program is stored in the storage specified in the "STORAGE" item.
When a existing LS file that does not have the item "STORAGE" is loaded, the program is stored in the storage specified in "Default Storage".

NOTE
"ASCII UPLOAD" option is necessary to load LS files.

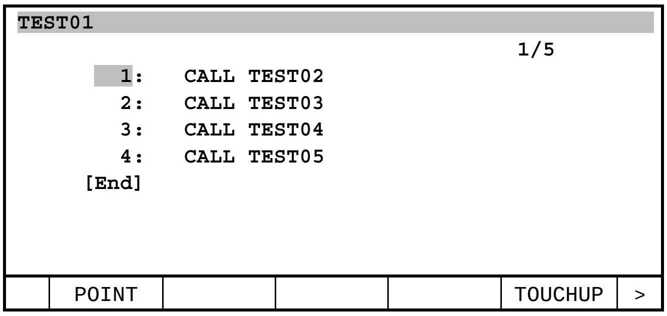
NOTE
"R709 TP DRAM/FILE Storage" option is necessary to load LS files that include "STORAGE" item. To load LS files into the controller without this option, you need to delete "STORAGE" item in advance.

29.3.4 Make Backup of Programs

In File screen, you can make backup of SHADOW and SHADOW ONDEMAND programs as well as CMOS programs.
However, when you make backup of FILE programs, you need to set "Backup FILE programs" "ON" in Program Configuration screen.

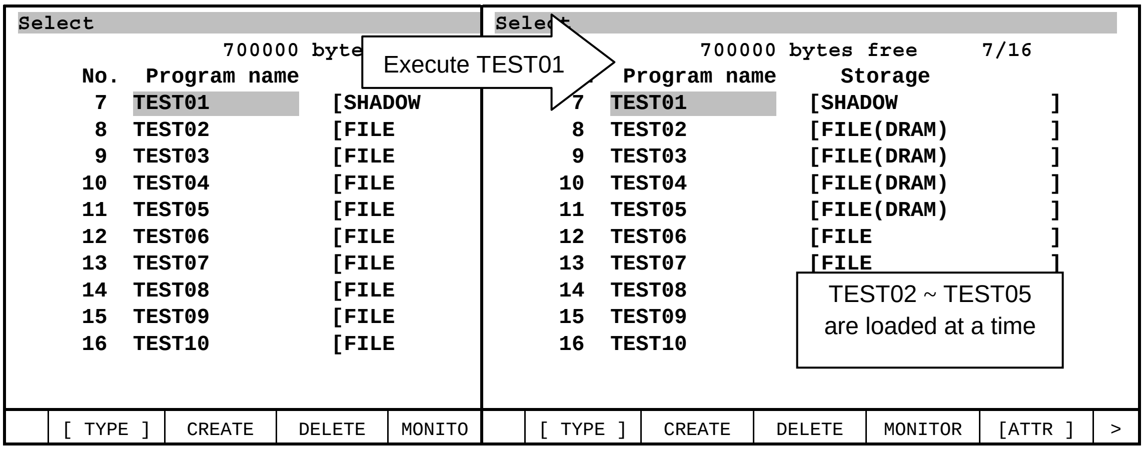
29.4 LOADING PROCESS IN PROGRAM EXECUTION

The programs that are not loaded on startup such as SHADOW ONDEMAND and FILE programs have time lag until they are loaded, because they are loaded on the start of edit or execution.
For example, SHADOW program TEST01 execute FILE programs TEST02~TEST05 by using CALL instruction.



```
TEST01
                                        1/5
      1:   CALL TEST02
      2:   CALL TEST03
      3:   CALL TEST04
      4:   CALL TEST05
   [End]
```

When TEST01 is executed, all the programs (TEST02~TEST05) that are called from TEST001 are loaded into DRAM.
If the size of a FILE program is large, it takes time to load it. So there is a time lag.
If these FILE programs are loaded in advance, TEST01 will start immediately.



```
Select
                700000 byte
     No.  Program name
      7   TEST01          [SHADOW
      8   TEST02          [FILE
      9   TEST03          [FILE
     10   TEST04          [FILE
     11   TEST05          [FILE
     12   TEST06          [FILE
     13   TEST07          [FILE
     14   TEST08          [FILE
     15   TEST09          [FILE
     16   TEST10          [FILE

   [ TYPE ]  CREATE  DELETE  MONITO
```

Execute TEST01

```
Select
                700000 bytes free     7/16
          Program name     Storage
      7   TEST01          [SHADOW            ]
      8   TEST02          [FILE(DRAM)        ]
      9   TEST03          [FILE(DRAM)        ]
     10   TEST04          [FILE(DRAM)        ]
     11   TEST05          [FILE(DRAM)        ]
     12   TEST06          [FILE              ]
     13   TEST07          [FILE              ]
     14   TEST08
     15   TEST09
     16   TEST10

   [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR ]  >
```

TEST02 ~ TEST05 are loaded at a time

29.5 PROGRAM EXCHANGE FUNCTION WITHOUT ENOUGH MEMORY SPACE

An error usually occurs when you load SHADOW ONDEMAND or FILE programs without enough memory space in DRAM. This function replaces old programs with newly loaded program in the same storage automatically.

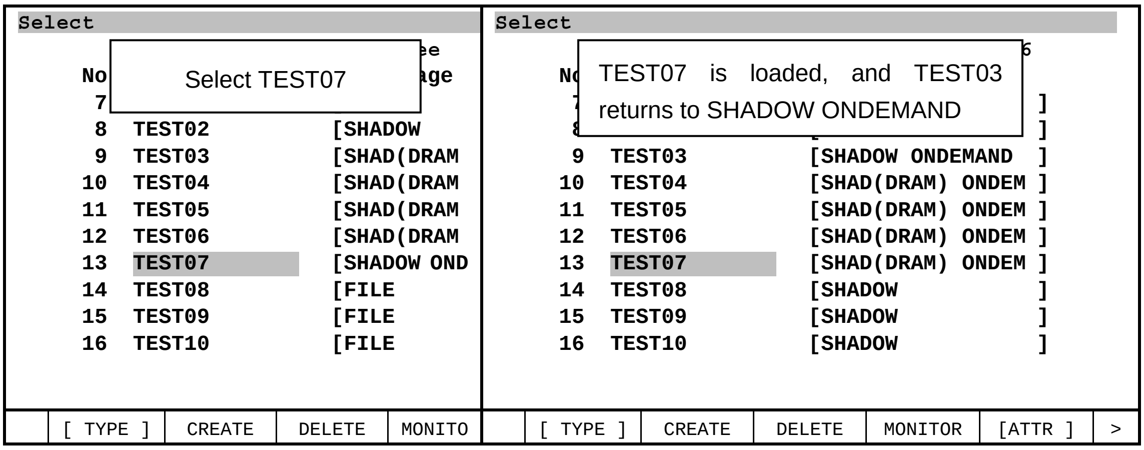
Until there is enough memory space in DRAM for the newly loaded program, old programs are deleted from DRAM in order automatically.

Old programs are deleted, but they still exist in the original storage. So they can be loaded again.

The following figure indicates the behavior of loading a new SHADOW ONDEMAND program, TEST07 when there is not enough memory space in DRAM. When TEST07 is selected, TEST03 moves back to SHADOW ONDEMAND and TEST07 can move into the ”temporary” storage state SHAD(DRAM) ONDE.

So, there is a time lag when you load TEST03 again.

You can use this function by setting the value of system variable $TPDRAM_CFG.$MANAGE_MEM to “1”. By default this value is “0” and this function is invalid.



```
Select
                                       ee
     No                                age
      7
      8   TEST02          [SHADOW
      9   TEST03          [SHAD(DRAM
     10   TEST04          [SHAD(DRAM
     11   TEST05          [SHAD(DRAM
     12   TEST06          [SHAD(DRAM
     13   TEST07          [SHADOW OND
     14   TEST08          [FILE
     15   TEST09          [FILE
     16   TEST10          [FILE

   [ TYPE ]  CREATE  DELETE  MONITO
```

Select TEST07

```
Select
                                          6
     N
      7                                    ]
      8                                    ]
      9   TEST03          [SHADOW ONDEMAND  ]
     10   TEST04          [SHAD(DRAM) ONDEM ]
     11   TEST05          [SHAD(DRAM) ONDEM ]
     12   TEST06          [SHAD(DRAM) ONDEM ]
     13   TEST07          [SHAD(DRAM) ONDEM ]
     14   TEST08          [SHADOW           ]
     15   TEST09          [SHADOW           ]
     16   TEST10          [SHADOW           ]

   [ TYPE ]  CREATE  DELETE  MONITOR  [ATTR ]  >
```

TEST07 is loaded, and TEST03 returns to SHADOW ONDEMAND

29.6 PRECAUTION

29.6.1 Cause and Remedy for Alarm Occurrence

Cause and remedy for alarms that occur when you use this function are listed.

TPIF-036 Not enough memory

Cause:

1 Performed such operations as creating a new program, edit, or copy a program without enough memory space in DRAM and CMOS.

2 Performed such operations as creating, editing, or copying a SHADOW, SHADOW ONDEMAND or FILE program in the following situation.
“Memory limit” is valid and the memory usage exceeds the limit.
There is not enough memory space in CMOS.

Remedy:

1 Delete unnecessary programs using CMOS and DRAM, and make enough memory space in CMOS and DRAM.

2 Set larger value in “Memory limit” or set the limit invalid. Delete unnecessary programs using CMOS and make enough memory space in CMOS.

MEMO-126 No more available TPP memory

Cause:

1 Performed such operations as editing, copying a program or creating a new CMOS program without enough memory space in CMOS.

Remedy:

1 Delete unnecessary programs using CMOS and make enough memory space in CMOS.

MEMO-178 No more available DRAM (SHADOW/FILE)

Cause:

1 Performed such operations as creating a new program, edit, or copy a program without enough memory space in DRAM.

2 Performed such operations as creating, editing, or copying a SHADOW, SHADOW ONDEMAND or FILE program in the following situation.
“Memory limit” is valid and the memory usage exceeds the limit.

Remedy:

1 Delete unnecessary programs using DRAM and make enough memory space in DRAM.

2 Set larger value in “Memory limit” or set the limit invalid. Delete unnecessary programs using DRAM and make enough memory space in DRAM.

TPIF-239 %s loaded to CMOS

Cause:

Performed such operations as loading a TP or LS program file of SHADOW, SHADOW ONDEMAND or FILE program in FILE screen in the following either situation.

1 Memory space in DRAM is not enough.

2 “Memory limit” is valid and the memory usage exceeds the limit.

Remedy:

1 Delete unnecessary programs using DRAM and make enough memory space in DRAM.

2 Set larger value in “Memory limit” or set the limit invalid. Delete unnecessary programs using DRAN and make enough memory space in DRAM.

After these operations, change storage individually form CMOS or load the program file again.

NOTE
When this alarm occurs, the program is loaded to CMOS.

30 CYCLE TIME LOGGING

Cycle Time Logging function records and displays the cycle time of programs. Cycle times are recorded based on individual cycles and hourly averages. Multiple cycles can be observed to view trends and analyze data. The cycle time data is displayed graphically and raw cycle data in text format can also be displayed. By default, the cycle time is tracked from the start of a teach pendant program until the program ends.
This is optional function.

Cycle Time Categories

Cycle Time is recorded based on multiple categories. The following table describes each category.

Table 30(a) Cycle time categories

Category	Description
MOTION	The amount of motion time in the cycle.
PROCESS	The amount of process time including welding time.
APPL	The amount of application-specific time executed such as ARC instruction.
WAIT	The amount of time spent waiting for I/O or a data event. Condition specified WAIT instruction is recorded.
DELAY	Time specified WAIT instruction is recorded.
KAREL	The amount of time spent executing KAREL logic.
LOGIC	The amount of time spent executing instructions except categories above (ex: Label, I/O).
HOMIO	The amount of time spent executing HOME_IO program. HOME_IO is automatically executed when the robot reaches the home position set in Reference Position screen and the program ran by SOP, UOP, or PROD_START ends. To execute HOME_IO, Reference position should be set correctly. If you do not use HOME_IO, home position setting is not necessary.
IDLE	The time, until the next cycle starts after the robot reaches home position and the program ends, is recorded.
FAULT	The amount of time spent during a program is paused by alarms or some other causes.
TOTAL	The total cycle time of categories above except HOMIO, IDLE and FAULT.

NOTE

The cycle time PROCESS, APPL, HOMIO, and IDLE is recorded only when the program is run by SOP cycle start, UOP_START, or PROD_START.

Cycle Time Screen Items

The following table describes the items displayed in STATUS Cycle Time screen.

Table 30(b) Items in cycle time screen

Item	Description
Program	The name of currently tracked program.
Mode	This item indicates whether the cycle is tracked per cycle or per hour. one, ten, or one hundred cycles can be displayed. It can also display one, ten, one hundred hours.
Date	The date of the currently highlighted cycle or hour recorded.
Time	The time of the currently highlighted cycle was executed.
Range (Total)	This indicates the total number of cycles as n/m where m is the total number of cycles or hours, and n is the currently displayed cycle or hour.
Samples	This item indicates the number of hourly samples used in the currently displayed average of hours.
[PROG]	This function key allows you to change the name of program for which cycle time information is displayed.
[ZOOM]	This function key allows you to change the displayed cycle time information. You can select to display one hour, ten hours, one hundred hours, one cycle, ten cycles, or one hundred cycles.

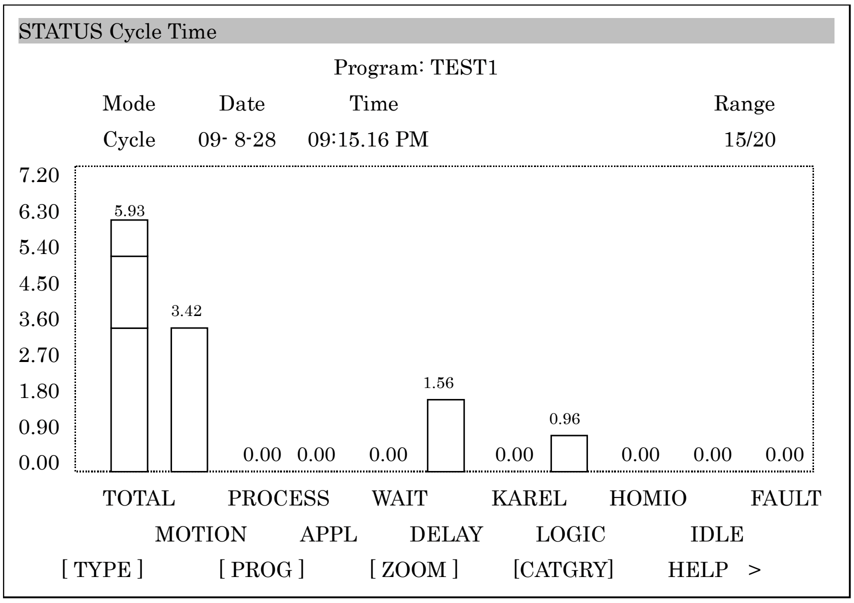
Item	Description
[CATGRY]	This function key allows you to choose which set of cycle time information is displayed. You can choose Total which displays the total of all of the categories and the cycle time for each individual category. Or, you can choose an individual category to be displayed. When an individual category is displayed, depending on the currently selected mode, the category name will be displayed on the top of right corner, and cycle time information for that category will be only displayed. You can choose to display the following categories: TOTAL, MOTION, PROCESS, APPL, WAIT, DELAY, KAREL, LOGIC, HOMIO, IDLE, FAULT.
[PAGE]	This function key allows you to display cycle time data in a variety of formats. To display cycle time data on the teach pendant, Cycle Graphics should be displayed. Other formats include Cycle Text, and Average Text. Line Text is only available in line-by-line mode. If you select Cycle Text or Average Text, the .DG files will be displayed. CYCLES.DG displays the cycle data, and CYCAVE.DG displays the hourly data. These files can be displayed using the Web Server on the teach pendant.
[UPDATE]	This function key is used for the 4 operations in the following. Each detail is described in 30.3.3. Load Cycle Data: Load the cycle time data. Clear Cycle Data: Clear the cycle time data. Clear Line Data: Clear the line data in line-by-line mode. Collect Line Data: Get the line data in line-by-line mode.
RECORD	This function key records the specific cycle time of a program. “RECORD” is displayed only in cycle mode when an individual category is selected. Detail is described in 30.3.5.

30.1 DISPLAY MODES OF CYCLE TIME

Cycle time data can be displayed in three modes by cycles, by hours, or line-by-line.

30.1.1 Cycle Mode

In cycle mode, cycle time is displayed by cycle. You can display cycle time data by one cycle, ten cycles, or one hundred cycles at a time. The following figure displays the total cycle time for the current cycle in one cycle mode. It also displays the total cycle time broken down by category. Each category is represented by a different color.

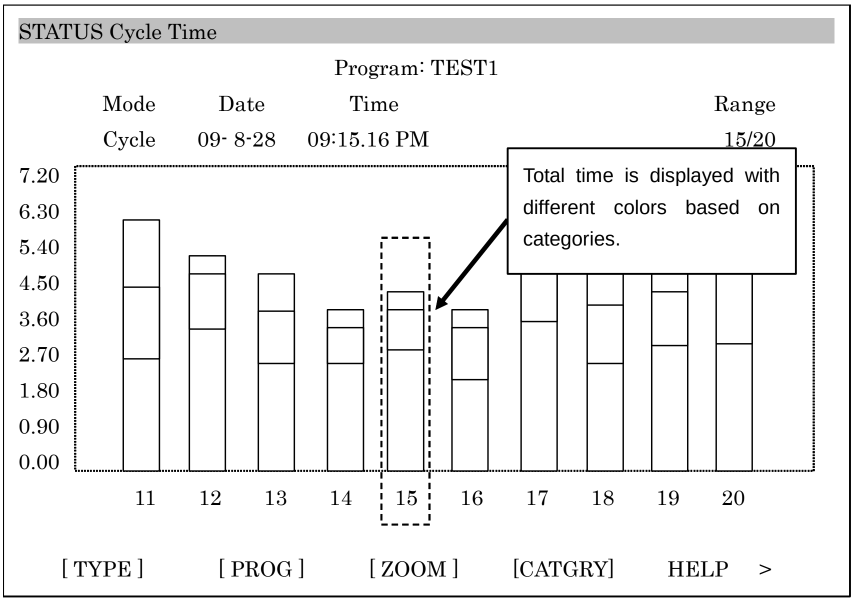


STATUS Cycle Time

Program: TEST1

Mode	Date	Time	Range
Cycle	09- 8-28	09:15.16 PM	15/20

The following figure displays cycle times in ten cycles mode.

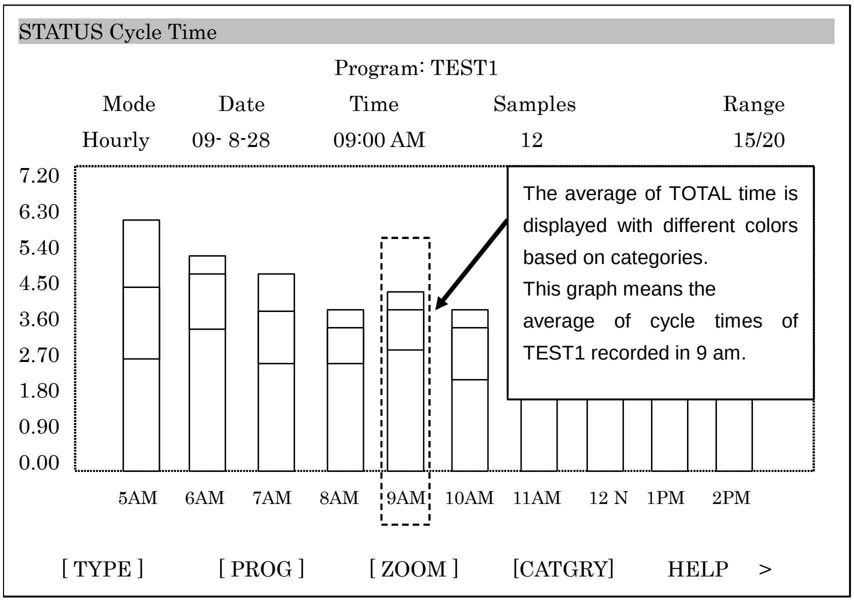


```
STATUS Cycle Time
                    Program: TEST1
   Mode     Date        Time                  Range
   Cycle    09- 8-28    09:15.16 PM           15/20
```

Total time is displayed with different colors based on categories.

30.1.2 Hourly Mode

In hourly mode, the hourly average cycle time is displayed. You can display cycle time data by one hour, ten hours, or one hundred hours at a time. Hourly mode is available only when hourly data exists.
The following figure displays the cycle time in ten hours mode.



```
STATUS Cycle Time
                    Program: TEST1
   Mode     Date        Time        Samples        Range
   Hourly   09- 8-28    09:00 AM      12           15/20
```

The average of TOTAL time is displayed with different colors based on categories.
This graph means the average of cycle times of TEST1 recorded in 9 am.

30.1.3 Line-by-Line Mode

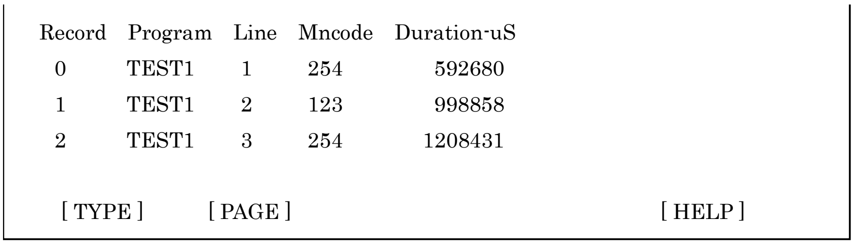
In line-by-line mode, the cycle time in each line in a program is recorded. So you can see the cycle time of specific line number in the program. The data is saved in CYCLINE.DG in DRAM. So it is cleared when the power is off.

To use line by line mode, $CY_CONFIG.$LINEENABLE and $CY_CONFIG.$NUMLINES should be configured.

$CY_CONFIG.$LINEENABLE	Valid when it is TRUE. (FALSE by default)
$CY_CONFIG.$NUMLINES	Set the number you want to record. (0 by default)

After that,

1. In STATUS Cycle Time screen, press NEXT, press F3, [UPDATE] and select Collect Line Data.
2. Execute a program of which you want the cycle time.
3. In STATUS Cycle Time screen, press NEXT, press F2, [PAGE] and select Line Text.
4. The cycle time of each line is displayed in the Web Server as follows.



```
Record  Program  Line  Mncode  Duration-uS
  0      TEST1    1     254        592680
  1      TEST1    2     123        998858
  2      TEST1    3     254       1208431

  [ TYPE ]     [ PAGE ]                          [ HELP ]
```

30.2 LOGGING CYCLE TIME

Procedure 30-1 Logging Cycle Time

Condition

- The program you want to track has been executed at least one time.

Step

1. Press [MENU] key.
2. Press STATUS.
3. Press F1, [TYPE].
4. Select Cycle Time.
5. Set the name of the program whose cycle time you want to display. By default, all programs are selected. To select a single program to display, press F2, [PROG], and select the name of the program.
6. To change the cycle time display between cycles and hours, press F3, [ZOOM] and select the kind of cycle time you want to display. After you have made a display selection (cycles or hours) you can quickly change the display by using the arrow keys. The up and down arrow keys move from one cycle or hour, to ten cycles or hours, then a hundred cycles or hours. The left and right arrow keys can move forward or backward through the cycles or hours. To move forward and backward quickly, press SHIFT and the left or right arrow key. If you are in ten cycle mode it will move by 5. If in one hundred cycle mode, it will move by 30.
7. To display cycle time for a specific program category, press F4, [CATGRY] and select the category you want to display.

30.2.1 Change Display Format of Cycle Time Data

The format of displaying cycle time data can be changed. In STATUS Cycle Time screen, press NEXT, press F2, [PAGE], and select one from the following items.

• Cycle Graphics	Cycle time data is displayed graphically.
• Cycle Text	Cycle time data is displayed as raw text. The data is saved in CYCLES.DG.
• Average Text	Hourly average cycle time is displayed as raw text. The data is saved in CYCAVE.DG.
• Line Text	Line data of cycle time in line-by-line mode is displayed as raw text. The data is saved in CYCLINE.DG and is cleared with power off. Line text can be displayed only when line-by-line mode is valid.

NOTE
Text data is displayed on Web Server screen.

30.2.2 Save Cycle Time Data

To save the cycle time data, press [FCTN] key and select SAVE. The cycle time data of cycle mode is saved in PGCYCCYC.TXT and that of hourly mode is saved in PGCYCHR.TXT. They are saved in the device selected in File screen.
F4, [BACKUP] in File screen can be also used.

30.2.3 Update Cycle Time Data

For update of cycle time data, there are 4 operations. In STATUS Cycle Time screen, press NEXT, press F3, [UPDATE], and select the operation you want to execute. The detail of each operation is described as follows.

- Load Cycle Data
  Clear the currently recorded cycle time data and load the previous cycle time data from the device that is selected in FILE screen.
  Loading PGCYCCY.TXT and PGCYCHR.TXT also can display the previous cycle time data.

- Clear Cycle Data
  Clear the currently recorded cycle time data.

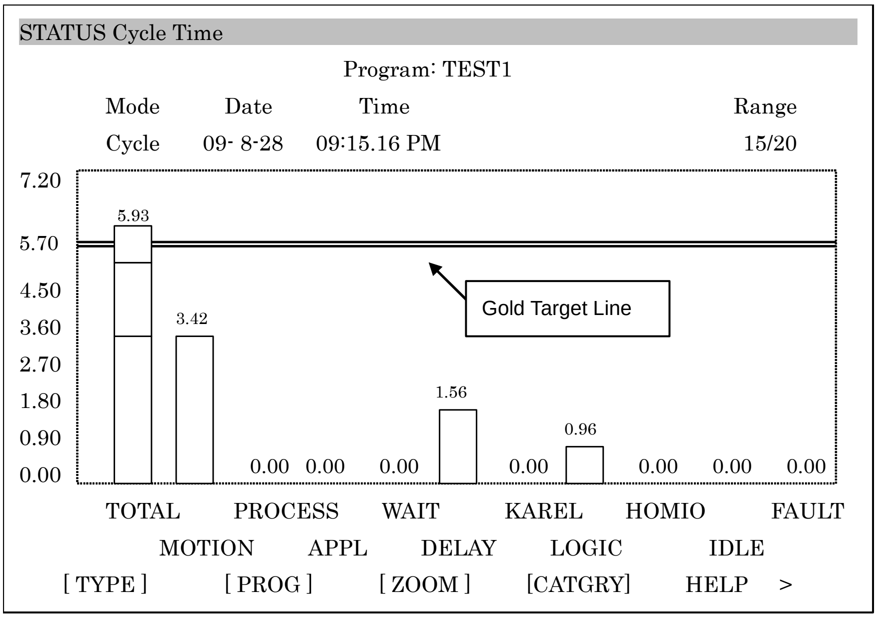
- Clear Line Data
  Clear the data for line by line mode. When you want to get a new line data, you should select Collect Line Data again.

- Collect Line Data
  Collect Line Data is an operation necessary for getting the line data for line-by-line mode. After this operation, execute a program whose cycle time you want to record. Then, in STATUS Cycle Time screen, press NEXT, press F2, [PAGE], and select Line Text. The cycle time for each line is displayed on Web Server screen.

30.2.4 Display Target Cycle Time Line

Target cycle time can be displayed as a gold line.
Target cycle time is displayed only in 1 cycle mode.

1. Press [SELECT] key and select a program.
2. Press [NEXT] key, and press F2, DETAIL.
3. Press F3, NEXT until Target Cycle Time (seconds) is displayed.
4. Press [ENTER] key and type a new target cycle time.
5. Press F1, END.
6. Press [MENU] key , select STATUS, press F1, [TYPE] and select Cycle Time if it is not already displayed.
7. Press F2, [PROG] and select the program for which target cycle time is set.
8. Press F3, [ZOOM] and select One Cycle. The following screen will be displayed.



STATUS Cycle Time

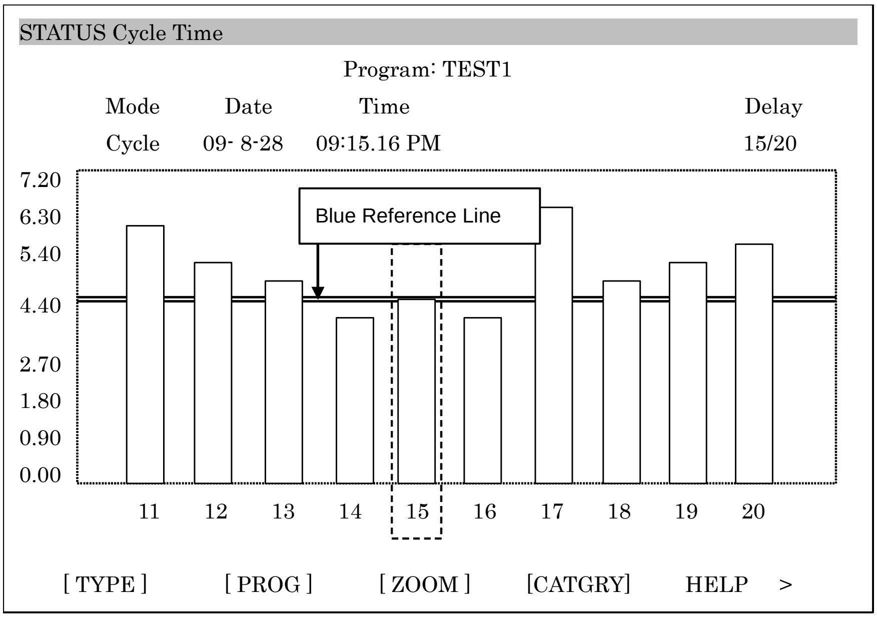
Program: TEST1

```
Mode     Date      Time               Range
Cycle    09- 8-28  09:15.16 PM        15/20
```

30.2.5 Display Reference Cycle Time Line

The cycle time of the current program can be recorded and be used as a reference. The recorded cycle time is displayed as a blue line. The reference cycle time line is displayed only in cycle mode and when an individual category is selected.

1. In STATUS Cycle Time screen, press F2, [PROG] and select the program whose cycle time you want to record.
2. Press NEXT, and confirm that F4, RECORD is displayed and press it.
3. You will be prompted to answer whether or not you want to overwrite. Select YES and the cycle time will be recorded.
4. Press F4, [CATGRY] and select a category you want to display. DELAY is selected in the following example.
5. The recorded cycle time is displayed as a blue line. (not displayed when TOTAL is selected.)



STATUS Cycle Time

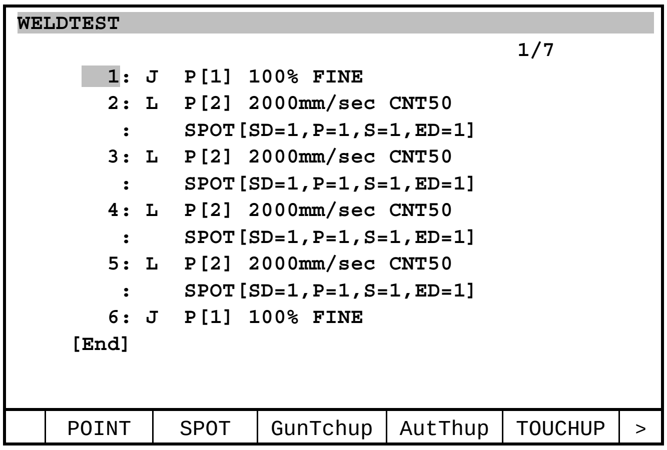
Program: TEST1

```
Mode     Date      Time               Delay
Cycle    09- 8-28  09:15.16 PM        15/20
```

30.3 EXECUTION EXAMPLE OF SPOT PROGRAM

In this section, as an execution example of Cycle Time Logging function, the cycle time is recorded by executing the following Spot program, WELDTEST.

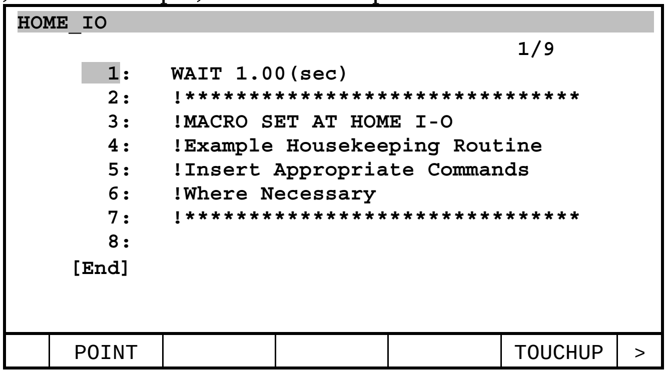
The cycle time of Spot program is not recorded by SHIFT-FWD execution. SOP cycle start, UOP_START, or PROD_START should be used.



```
WELDTEST
                                             1/7
      1: J  P[1] 100% FINE
      2: L  P[2] 2000mm/sec CNT50
       :    SPOT[SD=1,P=1,S=1,ED=1]
      3: L  P[2] 2000mm/sec CNT50
       :    SPOT[SD=1,P=1,S=1,ED=1]
      4: L  P[2] 2000mm/sec CNT50
       :    SPOT[SD=1,P=1,S=1,ED=1]
      5: L  P[2] 2000mm/sec CNT50
       :    SPOT[SD=1,P=1,S=1,ED=1]
      6: J  P[1] 100% FINE
    [End]
```

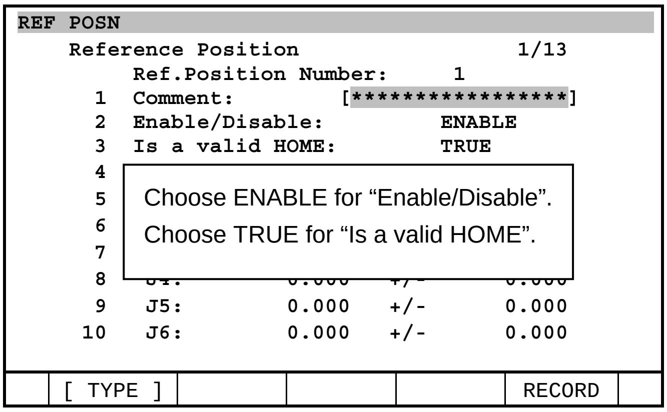
WELDTEST has 4 Spot instructions. In this example, the setting of welding time is 250 ms per one Spot instruction. So the total welding time is expected to 1 second and recorded as PROCESS cycle time.

The following HOME_IO program is also used. It has only one instruction of “WAIT 1.00 (sec)” and other statements are all comment. This is to make HOMIO cycle time easy to estimate. HOME_IO is automatically executed when the robot reaches the home position set in Reference Position screen and the program ran by SOP, UOP, or PROD_START ends. The execution time of HOME_IO is recorded as HOMIO cycle time. So, in this example, HOMIO is expected to be 1 second.



```
HOME_IO
                                             1/9
      1:   WAIT 1.00(sec)
      2:   !********************************
      3:   !MACRO SET AT HOME I-O
      4:   !Example Housekeeping Routine
      5:   !Insert Appropriate Commands
      6:   !Where Necessary
      7:   !********************************
      8:
    [End]
```

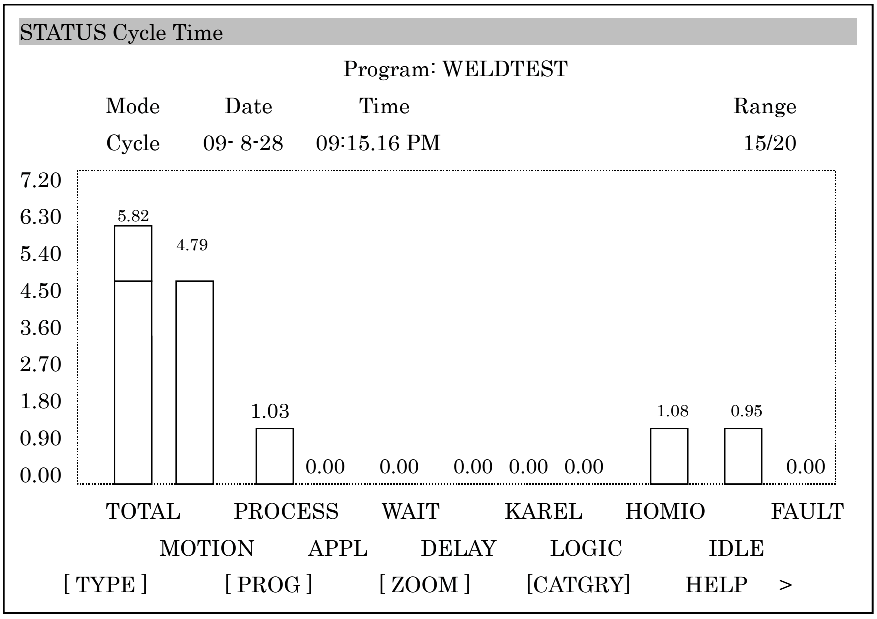
To execute HOME_IO, Reference position should be set correctly.



```
REF POSN
    Reference Position                   1/13
         Ref.Position Number:     1
     1   Comment:        [*****************]
     2   Enable/Disable:         ENABLE
     3   Is a valid HOME:        TRUE
     4
     5
     6
     7
     8    J4:         0.000   +/-      0.000
     9    J5:         0.000   +/-      0.000
    10    J6:         0.000   +/-      0.000
```

Choose ENABLE for “Enable/Disable”.
Choose TRUE for “Is a valid HOME”.

The following figure is the cycle time of WELDTEST.



STATUS Cycle Time

Program: WELDTEST

Mode	Date	Time	Range
Cycle	09- 8-28	09:15.16 PM	15/20

As expected, PROCESS cycle time is 1.03 second and HOMIO cycle time is 1.08 second.
Here, IDLE cycle time is recorded with the value 0.95 second. IDLE cycle time records the time that the robot stays idle until next program starts after the program ends. In this example, WELDTEST was restarted soon after it ended. This cycle time data shows that it took 0.95 second to restart WELDTEST after it ended.
The Total cycle time includes MOTION and PROCESS cycle times. HOMIO, IDLE, and FAULT are not included in TOTAL cycle time.

31 MATH FUNCTION INSTRUCTION

- Math function instruction enables to calculate math function (ex: SIN COS and so on) in the TP program.
- Math function instruction can be used in assignment statement, conditional statement and wait command.
- To use this function, Math function option (J593) is required.

31.1 TYPE OF MATH FUNCTIONS

Usable instructions of math functions are as follows.

Table 31.1 Usable instruction of math function

Function	Explanation	Restriction of argument
SQRT[x]	Square root	0 <= x
SIN[x]	Trigonometric function Units are degrees.	None
COS[x]		None
TAN[x]		Except 90,270±360ｎ
ASIN[x]		-1 <= x <= 1
ACOS[x]		-1 <= x <= 1
ATAN[x]		None
ATAN2[x,y]		Except x=0,y=0
LN[x]	Natural Logarithm	0 < x
EXP[x]	Exponent	None
ABS[x]	Absolute	None
TRUNC[x]	Truncate	-2.1*10^9 <= x <=2.1*10^9
ROUND[x]	Round off	-2.1*10^9 <= x <=2.1*10^9

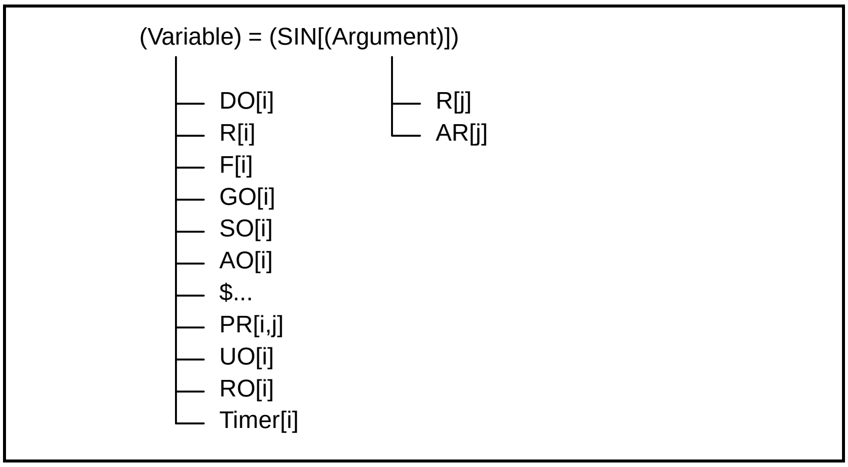
- Units of aforementioned functions are degrees.
- Conversion functions are not supported because the user can achieve them by multiplying a constant value (57.29579 or 0.017453 based on requirement).

31.2 INSTRUCTION FORMAT OF MATH FUNCTION

Instruction type of math functions is explained in the following example.

31.2.1 Instruction Format of Assignment Statements

- The function which only has one argument (Except of ATAN2[ ])

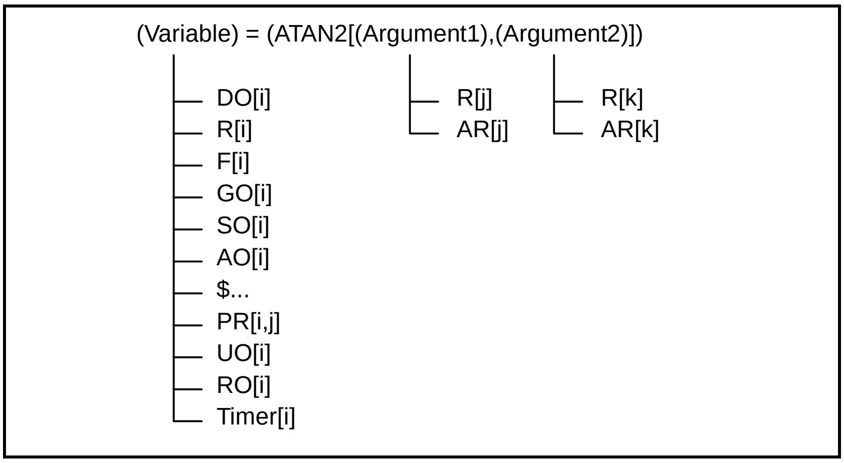


(Variable) = (SIN[(Argument)])

DO[i]
R[i]
F[i]
GO[i]
SO[i]
AO[i]
$...
PR[i,j]
UO[i]
RO[i]
Timer[i]

R[j]
AR[j]

- The function which has two arguments (ATAN2[ ])



(Variable) = (ATAN2[(Argument1),(Argument2)])

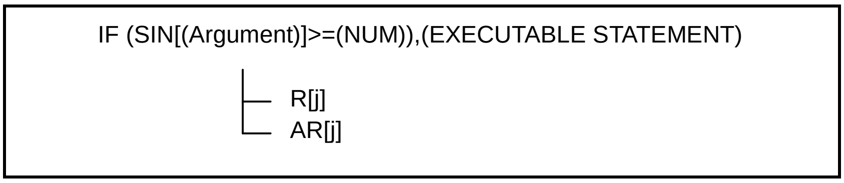
DO[i]
R[i]
F[i]
GO[i]
SO[i]
AO[i]
$...
PR[i,j]
UO[i]
RO[i]
Timer[i]

R[j]
AR[j]

R[k]
AR[k]

- The argument of math functions is “Register “ and “Argument Register”.
- Constant is not used directly.

31.2.2 Instruction Format of Relational Statements



IF (SIN[(Argument)]>=(NUM)),(EXECUTABLE STATEMENT)

R[j]
AR[j]

- A mixed logic instruction expression can be used in the conditional statement of a conditional branch command.
- If the result of conditional statement is ON, the executable statement of the conditional branch command is executed.
- The argument of math functions is “Register “ and “Argument Register”.
- Constant is not used directly.

31.2.3 Instruction Format of Wait Command Statements

```
WAIT (SIN[(Argument)]>=(NUM))

                    ├─ R[j]
                    └─ AR[j]
```

- A mixed logic expression can be specified in the conditional statement of a wait command.
- Wait until the specified condition is satisfied.
- The argument of math functions is “Register “ and “Argument Register”.
- Constant is not used directly.

31.3 FUNCTION SPECIFICATION OF MATH FUNCTIONS

31.3.1 Square Root（SQRT）

Function name: SQRT[X]
Argument: A positive integer or a positive real value to calculate square root.
This function calculates the positive square root of the argument.
Example:

```
1 : R[1] = 4
2 : R[2] = 2
3 : R[3] = (SQRT[R[1]])
4 : R[4] = (SQRT[R[2]])
```

Execute this TP program then,

```
R[3] = 2
R[4] = 1.41421
```

If the argument is minus value, then the following error occurs.

```
INTP-674        Invalid argument
```

31.3.2 Trigonometric Function（SIN）

Function name: SIN[X]
Argument: An angle to calculate SIN (in degree). An integer or a real value.
This function calculates sine of the argument.
Example:

```
1 : R[1] = 90
2 : R[2] = 45
3 : R[3] = (SIN[R[1]])
4 : R[4] = (SIN[R[2]])
```

Execute this TP program then,

```
R[3] = 1
R[4] = 0.70710
```

31.3.3 Trigonometric Function（COS）

Function name: COS[X]
Argument: An angle to calculate COS (in degree). An integer or a real value.
This function calculates cosine of the argument.
Example:

```
1 : R[1] = 90
2 : R[2] = 45
3 : R[3] = (COS[R[1]])
4 : R[4] = (COS[R[2]])
```

Execute this TP program then,

```
R[3] = 0
R[4] = 0.70710
```

31.3.4 Trigonometric Function（TAN）

Function name: TAN[X]
Argument: An angle to calculate TAN (in degree). An integer or a real value.
This function calculates tangent of the arguments.
Example:

```
1 : R[1] = 0
2 : R[2] = 45
3 : R[3] = (TAN[R[1]])
4 : R[4] = (TAN[R[2]])
```

Execute this TP program then,

```
R[3] = 0
R[4] = 1
```

If the argument equals to 180*n+90 and execute, then the following Error will be posted.

```
INTP-674          Invalid argument
```

31.3.5 Inverse Trigonometric Function（ASIN）

Function name: ASIN[x]
Argument: An integer or a real value to calculate ASIN. (Return value is degree.)
This function calculates arc sine of the argument.
Example:

```
1 : R[1] = 1
2 : R[2] = 0.5
3 : R[3] = (ASIN[R[1]])
4 : R[4] = (ASIN[R[2]])
```

Execute this TP program then,

```
R[3] = 90
R[4] = 30
```

If the argument ranges are x>1 or x<-1 and execute program, then the following Error will be posted.

```
INTP-674          Invalid argument
```

31.3.6 Inverse Trigonometric Function（ACOS）

Function name: ACOS[x]
Argument: An integer or a real value to calculate ACOS. (Return value is degree.)
This function calculates arc cosine of the argument.
Example:

```
1 : R[1] = 1
2 : R[2] = 0.5
3 : R[3] = (ACOS[R[1]])
4 : R[4] = (ACOS[R[2]])
```

Execute this TP program then,

```
R[3] = 0
R[4] = 60
```

If the argument ranges are x>1 or x<-1 and execute program, then the following Error will be posted.

```
INTP-674          Invalid argument
```

31.3.7 Inverse Trigonometric Function（ATAN2）

Function name: ATAN2[x,y]
1st argument: x coordinates of the point to calculate ATAN2. An integer or a real value.
2nd argument: y coordinates of the point to calculate ATAN2. An integer or a real value.
This function calculates arc tangent of the 1st and 2nd arguments. (Return value is degree.)
Example:

```
1 : R[1] = -1
2 : R[2] = 0.5
3 : R[3] = (ATAN2[R[1],R[2]])
4 : R[4] = (ATAN2[R[2],R[1]])
```

Execute this TP program then,

```
R[3] = 153.4349
R[4] = -63.43
```

31.3.8 Inverse Trigonometric Function（ATAN）

Function name: ATAN [x]
Argument: An integer or a real value to calculate ATAN.
This function calculates arc tangent of the argument. (Return value is degree.)
Example:

```
1 : R[1] = 1
2 : R[2] = 0.5
3 : R[3] = (ATAN[R[1]])
4 : R[4] = (ATAN[R[2]])
```

Execute this TP program then,

```
R[3] = 45
R[4] = 26.56
```

31.3.9 Exponent

Function name: EXP[x]
Argument: An integer or a real value to calculate EXP.
This function calculates a value equal to e (approximately 2.71828) raised to the power specified by the argument.
Example:

```
1 : R[1] = 0
2 : R[2] = 1
3 : R[3] = (EXP[R[1]])
4 : R[4] = (EXP[R[2]])
```

Execute this TP program then,

```
R[3] = 1
R[4] = 2.71828
```

31.3.10 Natural Logarithm

Function name: LN[x]
Argument: An integer or a real value to calculate LN.
This function calculates the natural logarithm of the argument.
Example:

```
1 : R[1] = 1
2 : R[2] = 2.718281828
3 : R[3] = (LN[R[1]])
4 : R[4] = (LN[R[2]])
```

Execute this TP program then,

```
R[3] = 0
R[4] = 1
```

If the argument is x <= 0 and the program execute, then the following Error will be posted.

```
INTP-674        Invalid argument
```

31.3.11 Absolute（ABS）

Function name: ABS[x]
Argument: An integer or a real value to calculate ABS.
This function calculates the absolute value of the argument.
Example:

```
1 : R[1] = 5.5
2 : R[2] = -5.2
3 : R[3] = (ABS[R[1]])
4 : R[4] = (ABS[R[2]])
```

Execute this TP program then,

```
R[3] = 5.5
R[4] = 5.2
```

31.3.12 Truncate（TRUNC）

Function name: TRUNC[x]
Argument: A real value to calculate TRUNC.
This function converts the real argument to an integer by removing fractional part of the real value.
Example:

```
1 : R[1] = 5.5
2 : R[2] = -5.2
3 : R[3] = (TRUNC[R[1]])
4 : R[4] = (TRUNC[R[2]])
```

Execute this TP program then,

```
R[3] = 5
R[4] = -5
```

31.3.13 Round Off（ROUND）

Function name: ROUND[x]
Argument: A real value to calculate ROUND.
This function calculates integer value closest to the real argument.
Example:

```
1 : R[1] = 5.5
2 : R[2] = -5.2
3 : R[3] = (ROUND[R[1]])
4 : R[4] = (ROUND[R[2]])
```

Execute this TP program then,

```
R[3] = 6
R[4] = -5
```

31.4 BACKGROUND OPERATION OF MATH FUNCTION

Math function instructions are part of the mixed logic instruction.
Then Math function instruction can be used in background operation.

Table 31.4 Background operation of math function

Mode	Maximum allowable number of items	Scanning time	Available data	Available operators
Standard mode	No restriction	(Number of items in standard mode / 300-Number of items in high-level mode)*ITP The number of items as used here refers to the total number of items in each mode background operation programs. One ITP is usually eight milliseconds.	SIN, COS, TAN, ASIN, ACOS, ATAN, ATAN2, SQRT, LN, EXP, ABS, TRUNC, ROUND	(,), =, <>, <, <=, >, >=, +, -, *, /, DIV, MOD
High-level mode	270	Eight milliseconds.		

- The scanning time of Math function is 2 or 3 times of normal items, because Math function instruction needs more time to calculate.
- The scanning time of ATAN2 instruction is 3 times of normal items.
- The other math function instructions of scanning time are 2 times of normal items.

31.5 TEACH MATH FUNCTION INSTRUCTION

Math function can be used in mixed logic instruction.
As an example, the teaching process of following statement is as follows.

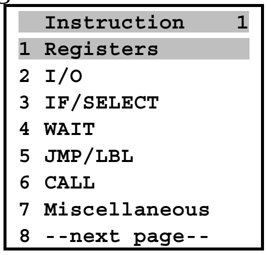
```
1 : R[1] = (SIN[R[2]])
```

Procedure 31-1 Examples of teaching math function

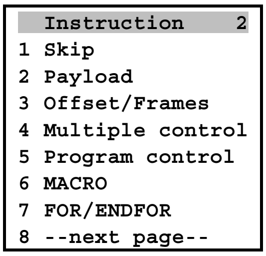
Step

1 Press F1, [INST] key.

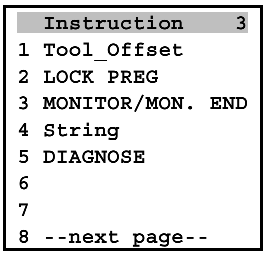
2 Select Register.



```
  Instruction    1
1 Registers
2 I/O
3 IF/SELECT
4 WAIT
5 JMP/LBL
6 CALL
7 Miscellaneous
8 --next page--
```

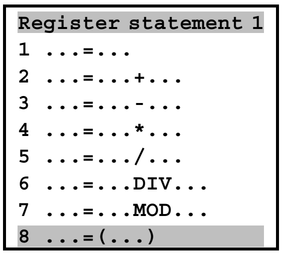


```
  Instruction    2
1 Skip
2 Payload
3 Offset/Frames
4 Multiple control
5 Program control
6 MACRO
7 FOR/ENDFOR
8 --next page--
```



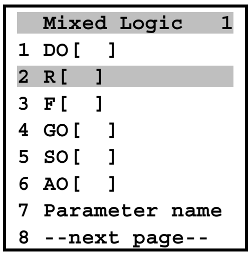
```
  Instruction    3
1 Tool_Offset
2 LOCK PREG
3 MONITOR/MON. END
4 String
5 DIAGNOSE
6
7
8 --next page--
```

3 Select mixed logic instructions "…= (…)".

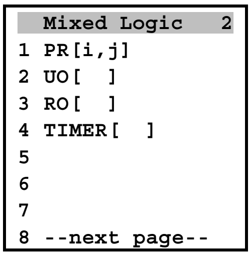


```
Register statement 1
1 ...=...
2 ...=...+...
3 ...=...-...
4 ...=...*...
5 ...=.../...
6 ...=...DIV...
7 ...=...MOD...
8 ...=(...)
```

4 Select register "R[ ]". Then the cursor moves in the square bracket.

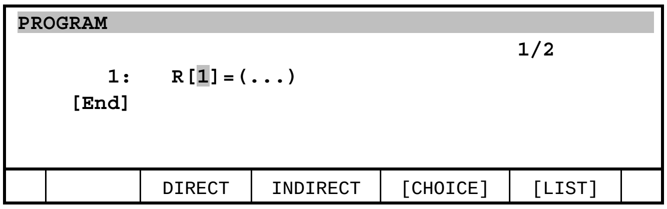


```
 Mixed Logic   1
1 DO[  ]
2 R[  ]
3 F[  ]
4 GO[  ]
5 SO[  ]
6 AO[  ]
7 Parameter name
8 --next page--
```



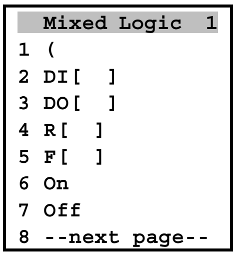
```
 Mixed Logic   2
1 PR[i,j]
2 UO[  ]
3 RO[  ]
4 TIMER[  ]
5
6
7
8 --next page--
```

5 Input index 1.

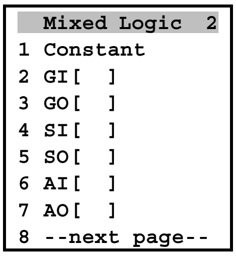


```
PROGRAM
                                        1/2
    1:   R[1]=(...)
  [End]

        DIRECT  INDIRECT  [CHOICE]  [LIST]
```

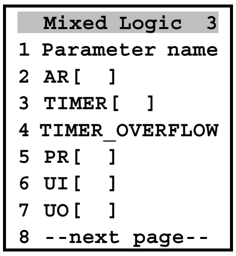
6 Then the cursor moves to the right side of equal sign.
7 Select "SIN[ ]".



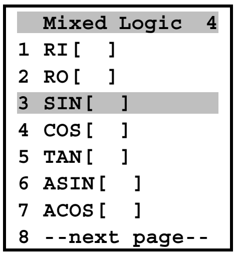
```
 Mixed Logic  1
1 (
2 DI[  ]
3 DO[  ]
4 R[  ]
5 F[  ]
6 On
7 Off
8 --next page--
```



```
 Mixed Logic  2
1 Constant
2 GI[  ]
3 GO[  ]
4 SI[  ]
5 SO[  ]
6 AI[  ]
7 AO[  ]
8 --next page--
```

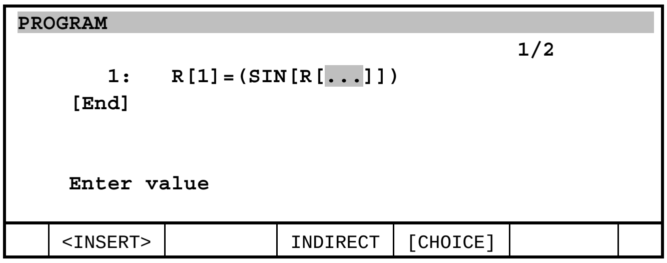


```
 Mixed Logic  3
1 Parameter name
2 AR[  ]
3 TIMER[  ]
4 TIMER_OVERFLOW
5 PR[  ]
6 UI[  ]
7 UO[  ]
8 --next page--
```



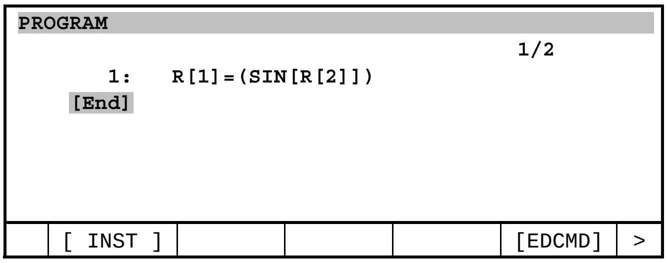
```
 Mixed Logic  4
1 RI[  ]
2 RO[  ]
3 SIN[  ]
4 COS[  ]
5 TAN[  ]
6 ASIN[  ]
7 ACOS[  ]
8 --next page--
```

8 Then the cursor moves in the square bracket. And the default value R[…] is selected.
Arguments are "Register " and "Argument Register". (Not Constant)
The "Register" is default value of argument.
So to use "Argument Register", press F3, [CHOICE] then select "Argument Register".
9 For example Register[2] is used here. Input value 2 then the cursor moves to the right parentheses.



```
PROGRAM
                                        1/2
    1:   R[1]=(SIN[R[...]])
  [End]


  Enter value

  <INSERT>          INDIRECT  [CHOICE]
```

10 Move cursor to finish teaching.
11 Teaching of SIN[R[2]] function completed.



```
PROGRAM
                                              1/2
     1:   R[1]=(SIN[R[2]])
   [End]

   [ INST ]                           [EDCMD]   >
```

31.6 RESTRICTION OF TEACHING MATH FUNCTION

- The argument of math functions is “Register “ and “Argument Register”.
- Constant is not used directly.
- If the user wants to use constant, input constant to a register, and use the register.
- You can put multiple math functions in a single line.
- You cannot put math functions in a math function.

Following statements are supported.

```
1 : R[1] = (SIN[R[2]])
2 : R[2] = (COS[AR[3]])
3 : R[3] = (SIN[R[4]] + COS[R]4)])
```

Following statements are not supported.

```
R[1] = (SIN[90])
        * Input constant directly.
R[3] = (SQRT[SIN[R[4]] * SIN[R[4]] + COS[R[4]]*COS[R[4]]])
        * SIN and COS are in SQRT.
```

- Math function might have a negligible calculation error of 10^{-7} due to internal calculation error because math function follows the same specification of KAREL Built-Ins.
- For example the result of COS[R[1]] (when R[1]=90), may be 10^{-8} that is not exactly zero. This behavior can be avoided by rounding off the result using ROUND function.

31.7 EXCEPTIONS AND RESTRICTION

Math function instruction posts the alarm in the following conditions.

- When the argument exceeds the domain of function definition.

Example	:LN[R[1]]	(R[1] = -1)
	:ASIN[R[2]]	(R[2] = 10)
	:SQRT[R[3]]	(R[3] = -4)
	:TAN[R[4]]	(R[4] = 90)

```
INTP-674      Invalid argument
```

- The value overflows when executed.

Example	:EXP[R[1]]	(R[1] = 100)

```
INTP-323      Value overflow
```

- If the variables are divided by zero then, the following error will be occurred.
  Example :R[1] / SIN[R[2]] (R[2] = 0)
  :10/COS[R[3]] (R[3] = 90)

```
INTP-208          Divide by 0
```

32 SERVO TOOL CHANGE FUNCTION

32.1 OUTLINE

32.1.1 Feature of Function

- It is possible to switch more than two servo tools without power off.
- Switching the air drive tool and the servo tool is also available.

What is a servo tool?

A servo tool is a tool having an axis that is driven by a servo motor controlled by a robot controller.

32.1.2 Basic Specification

The total number of changeable tool	4 for one-axis servo tool change 2 for two-axis servo tool change 1 for three-axis servo tool change Only one tool can be controlled at the same time.
The period of tool detach (standard)	0.4 sec (Battery type) / 0.4 sec(Battery less type) *1)
The period of tool attach (standard)	4 sec(Battery type) / 6 sec(Batteryless type) *1)
Backup battery	Both battery type and batteryless type can be available.*2)

(*1) The period includes the internal processing period, not include the period of robot motion for changing the tools.

(*2) The batteryless type is not available for multi-axis servo tool change.

32.1.3 Restrictions

Restrictions matters

- Switching is possible only between servo tools that are set within the same motion group.
- Because servo tools that do not have a battery for tool axis pulse value backup require calibration after they are attached, these tools require longer processing time for tool change than battery-equipped tools.
- The motor of the tool change must be controlled by the same amp.
- The tool change instruction can not be used by multi task. (The tool change instruction can not be executed by plural tasks at the same time.)
- Installing both the servo tool change function and servo gun change function cannot be done.
- On a system with the servo tool change function installed, the dynamic brake release function cannot be used for servo guns.
- The motion group that the tool change function is enabled must be used a final group in the system.

Prohibition matters

- Do not force to detach the tool while the controller is off.
- The line tracking can not be used with the tool change function.

Notes on using the batteryless type

- When calibration motion type 1 or 2 is selected, calibration cannot be interrupted by a hold or emergency stop. (Program execution must not be stopped during tool attach processing.)
- When calibration motion type 3 or 4 is selected, the stroke limit must be expanded so that the tool axis touches the open limit.

- When calibration motion type 5 or 6 is selected, a detection switch such as a limit switch must be installed in the tool.

For details of the calibration motion types, see Section 32.4, "TOOL CHANGE SETUP".

Notes on using multi-axis servo tool

- The servo tool number must be assigned to successive axes.
- The servo tool must have the backup battery.
- To change two-axis servo tool, the two-axis tool change unit must be used.
- To change three-axis servo tool, the three-axis tool change unit must be used.
- Four (or more)-axis servo tool cannot be changed.
- The servo tool cannot be changed to the one with a different number of axes.

NOTE

When a Pulsecoder *αi*AR128(A860-2010-T341) shipped until May of 2006 is used, alarm SRVO-068(DTERR) occurs at turning power on and executing TOOL ATTACH instruction. Implement either of following countermeasures.

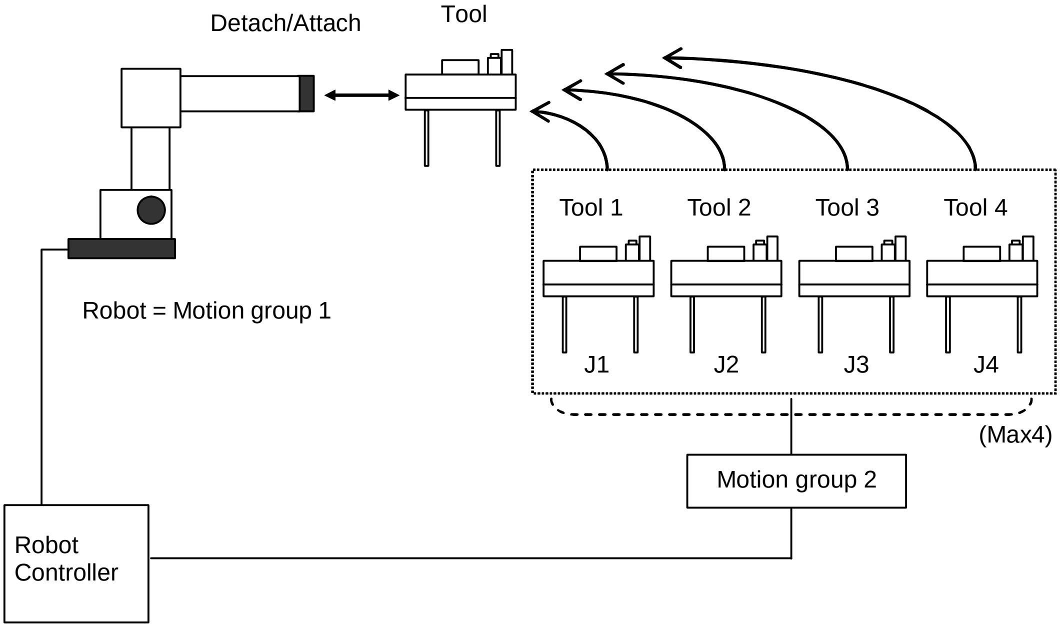
1. Replace Pulsecoder *αi*AR128 with one shipped after June of 2006.
2. Change system valuables as follows.
    - $SHC_ITF[#].$WAITTM_PCHK = 270
    - $SHC_ITF[#].$WAITTM_PWUP = 550
    - $SHC_ITF[#].$SVON_DELAY = 230
    - $SHC_ITF[#].$WAITTM_STRT = 650

    # is group number of servo tool group using Servo Tool Change Function.

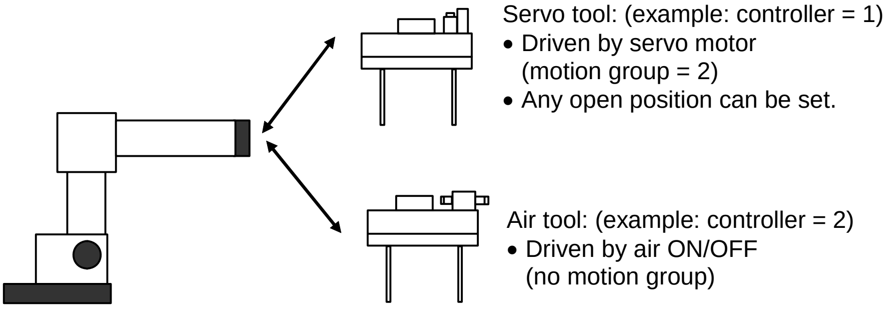
When you implement 2nd countermeasure, tool attach process is extended by about 2 seconds.

32.1.4 System Configuration

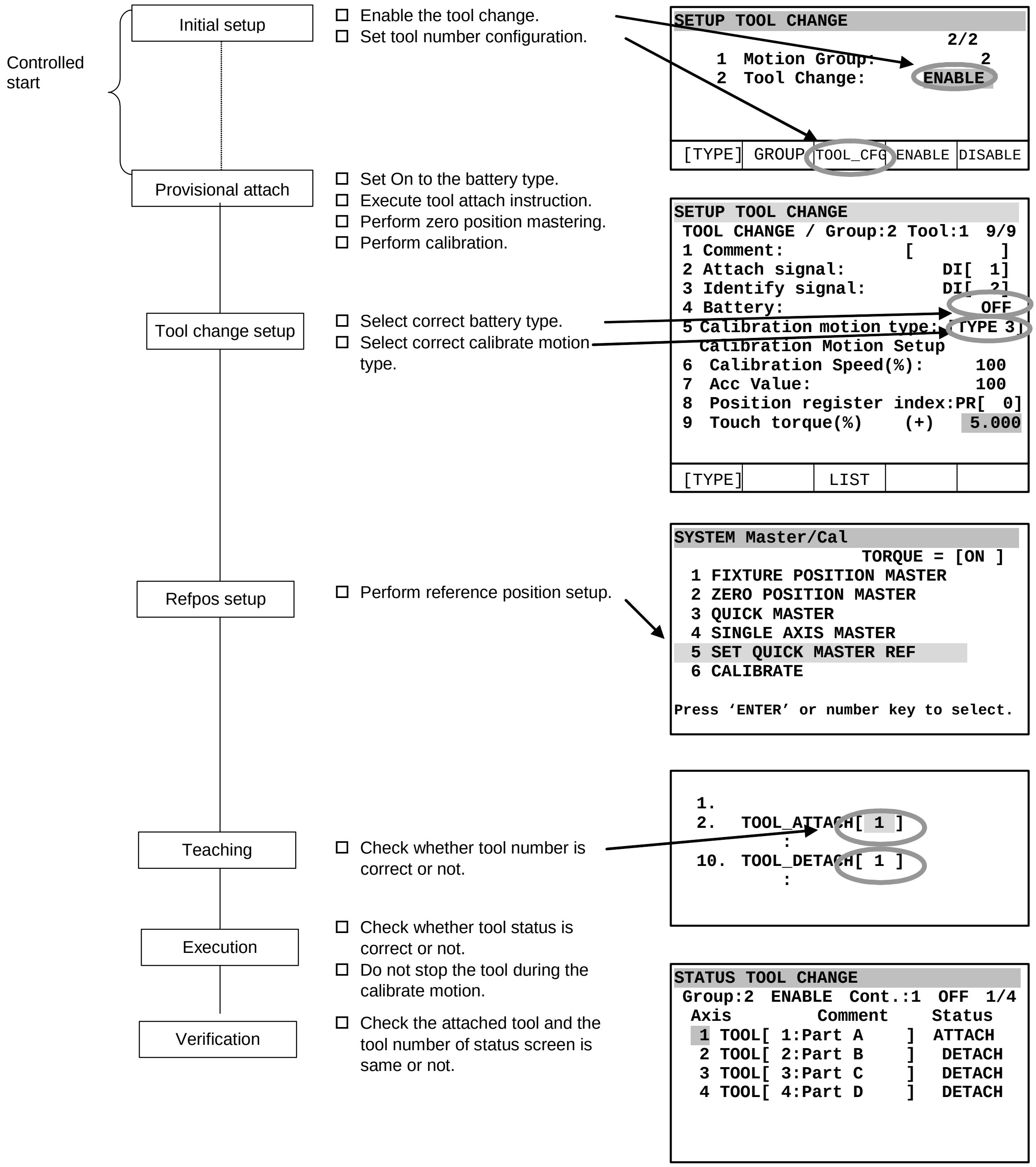
One-axis servo tool change system is constructed as follows.
(Example: Servo tool is the 2nd group and the number of tools is four.)



You can replace the servo tool with the air tool by switching on the controller.


Servo tool: (example: controller = 1)
• Driven by servo motor
(motion group = 2)
• Any open position can be set.
Air tool: (example: controller = 2)
• Driven by air ON/OFF
(no motion group)

32.1.5 Outline of Installation



Initial setup

- □ Enable the tool change.
- □ Set tool number configuration.

```
SETUP TOOL CHANGE
                              2/2
    1  Motion Group:           2
    2  Tool Change:        ENABLE

 [TYPE]  GROUP  TOOL_CFG  ENABLE  DISABLE
```

Provisional attach

- □ Set On to the battery type.
- □ Execute tool attach instruction.
- □ Perform zero position mastering.
- □ Perform calibration.

Tool change setup

- □ Select correct battery type.
- □ Select correct calibrate motion type.

```
SETUP TOOL CHANGE
 TOOL CHANGE / Group:2 Tool:1  9/9
 1 Comment:              [        ]
 2 Attach signal:           DI[  1]
 3 Identify signal:         DI[  2]
 4 Battery:                     OFF
 5 Calibration motion type: [TYPE 3]
   Calibration Motion Setup
 6  Calibration Speed(%):      100
 7  Acc Value:                 100
 8  Position register index:PR[  0]
 9  Touch torque(%)    (+)    5.000

 [TYPE]         LIST
```

Refpos setup

- □ Perform reference position setup.

```
SYSTEM Master/Cal
                     TORQUE = [ON ]
  1 FIXTURE POSITION MASTER
  2 ZERO POSITION MASTER
  3 QUICK MASTER
  4 SINGLE AXIS MASTER
  5 SET QUICK MASTER REF
  6 CALIBRATE

Press 'ENTER' or number key to select.
```

Teaching

- □ Check whether tool number is correct or not.

```
  1.
  2.   TOOL_ATTACH[ 1 ]
             :
 10.   TOOL_DETACH[ 1 ]
             :
```

Execution

- □ Check whether tool status is correct or not.
- □ Do not stop the tool during the calibrate motion.

Verification

- □ Check the attached tool and the tool number of status screen is same or not.

```
STATUS TOOL CHANGE
 Group:2  ENABLE  Cont.:1  OFF  1/4
  Axis         Comment        Status
   1 TOOL[ 1:Part A     ]  ATTACH
   2 TOOL[ 2:Part B     ]   DETACH
   3 TOOL[ 3:Part C     ]   DETACH
   4 TOOL[ 4:Part D     ]   DETACH
```

32.2 INITIAL SETUP

The following initial setup is needed to install the servo tool change system.

- Motion parameter setup (Independent axis setup) of servo tool axes
- The configuration setup of servo tool number
- Enable the tool change function.

> **NOTE**
> These setup must be done at the beginning of system installation. These setup must be done only one time.

These setup is done at the controlled start. See Subsections 32.11.1, 32.11.2 and 32.11.3 for detailed information.

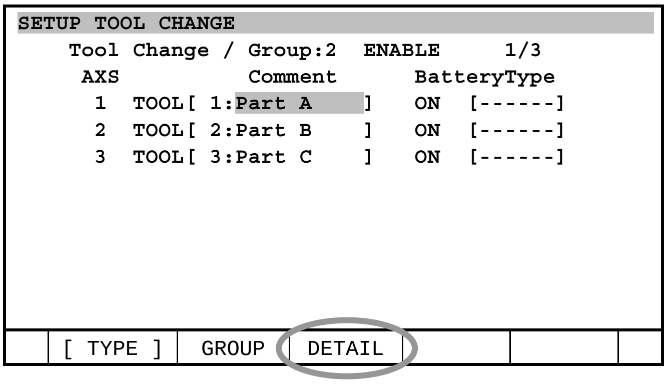
32.3 PRELIMINARY TOOL ATTACH OPERATION

When “Tool Change” is enabled as described in Section 32.2, "INITIAL SETUP", tools assigned to a target operation group are all detached.

> **NOTE**
> Only when the tool change software is installed in a system in which servo tools have already been operated and a tool used previously is left attached, the tool is attached. In this case, preliminary tool attach steps 1 to 9 explained below are unnecessary.

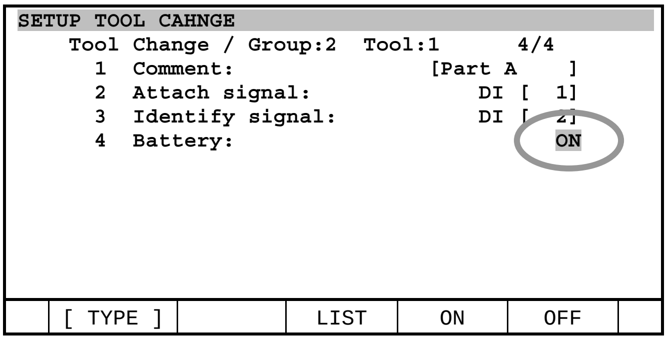
When using a tool for the first time, be sure to follow the steps below to perform the preliminary attach operation, then proceed to Section 32.4, "TOOL CHANGE SETUP".

1 Press [MENU] key to display the screen menu.
2 Select 6, SETUP.
3 Press F1, [TYPE] to display the screen switch menu.
4 Select “Tool Change”. The tool change list screen will be displayed.
5 From the list, select one tool, and press F3, DETAIL. The tool change setup detail screen will be displayed.



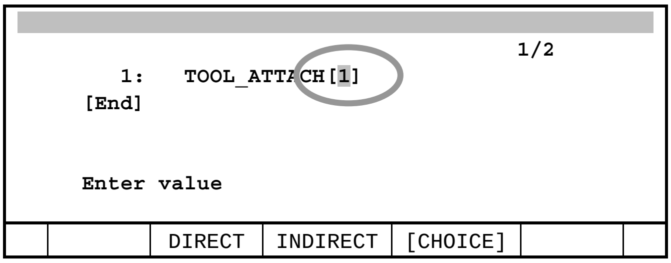
```
SETUP TOOL CHANGE
    Tool Change / Group:2  ENABLE     1/3
     AXS          Comment       BatteryType
      1  TOOL[ 1:Part A    ]    ON  [------]
      2  TOOL[ 2:Part B    ]    ON  [------]
      3  TOOL[ 3:Part C    ]    ON  [------]
```

6 On the tool change setup detail screen, temporarily set the battery to ON. (Even when the actual tool has no battery, select ON in this step. This item is to be set again after the preliminary attach operation is completed.)



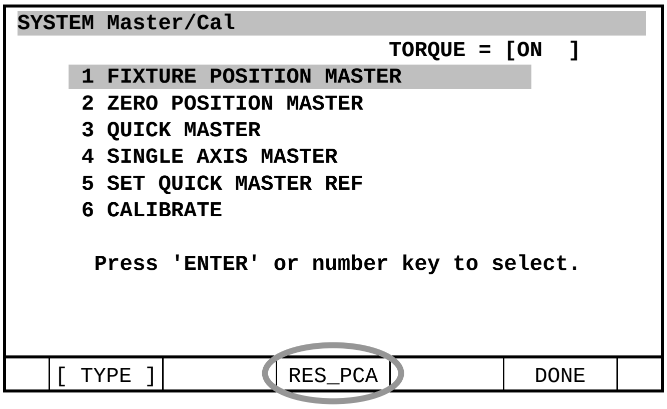
```
SETUP TOOL CAHNGE
    Tool Change / Group:2  Tool:1      4/4
      1  Comment:                [Part A    ]
      2  Attach signal:              DI [  1]
      3  Identify signal:            DI [  2]
      4  Battery:                          ON
```

7 Attach a target tool to the tool changer.
8 Create a program to teach an attach instruction.
9 Execute the TOOL ATTACH instruction to attach the tool temporarily.



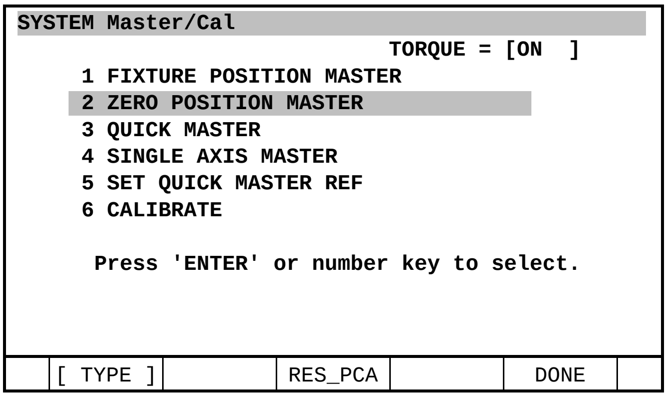
```
                                                1/2
     1:   TOOL_ATTACH[1]
  [End]


  Enter value
```

10 After the TOOL ATTACH instruction is executed, the following are issued:
SRVO-075 Pulse not established
SRVO-062 BZAL alarm
11 On the calibration screen, press F3, RES_PCA, and select YES.



```
SYSTEM Master/Cal
                              TORQUE = [ON  ]
    1 FIXTURE POSITION MASTER
    2 ZERO POSITION MASTER
    3 QUICK MASTER
    4 SINGLE AXIS MASTER
    5 SET QUICK MASTER REF
    6 CALIBRATE

     Press 'ENTER' or number key to select.
```

12 Cycle power of the controller.
13 Jog the servo tool axis.
Continue jogging the servo tool axis until the following message is displayed when the reset button is pressed:
SRVO-075 Pulse not established.
14 In the same manner as for ordinary servo tools, perform zero-position mastering for the servo tool axis.

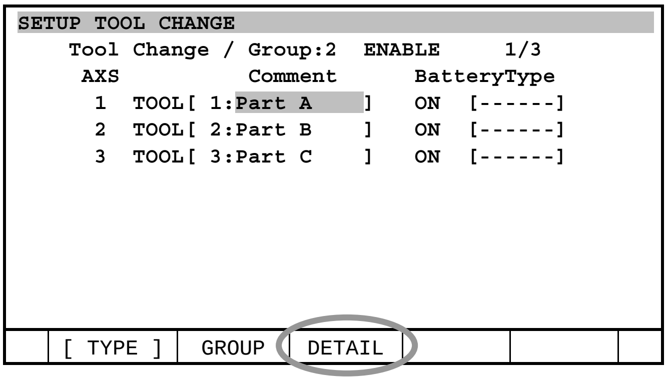


```
SYSTEM Master/Cal
                              TORQUE = [ON  ]
    1 FIXTURE POSITION MASTER
    2 ZERO POSITION MASTER
    3 QUICK MASTER
    4 SINGLE AXIS MASTER
    5 SET QUICK MASTER REF
    6 CALIBRATE

     Press 'ENTER' or number key to select.
```

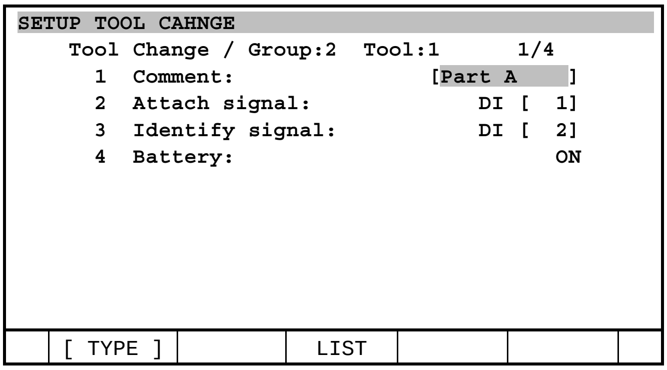
15 After completing tool change setup and reference position setting for the first tool, detach the first tool by using the TOOL DETACH instruction, then perform the preliminary attach operation for the second and subsequent tools.

32.4 TOOL CHANGE SETUP

The tool change setup must be done for each tool.



```
SETUP TOOL CHANGE
    Tool Change / Group:2  ENABLE     1/3
     AXS          Comment      BatteryType
      1  TOOL[ 1:Part A    ]   ON  [------]
      2  TOOL[ 2:Part B    ]   ON  [------]
      3  TOOL[ 3:Part C    ]   ON  [------]
```



```
SETUP TOOL CAHNGE
    Tool Change / Group:2  Tool:1      1/4
      1  Comment:               [Part A    ]
      2  Attach signal:             DI [  1]
      3  Identify signal:           DI [  2]
      4  Battery:                         ON
```

1 Press [MENU] key.
2 Select “6 SETUP”.
3 Press F1, [TYPE].
4 Select “Tool change”.
5 Select tool from the list of tools, then press F3, DETAIL. The detail screen will be displayed.

Common setup

Comment	Comment for each tool.
Attach signal	This signal used for tool set verification. Available I/O type = DI / RI • This signal is check before tool attach/detach instruction. If this signal is OFF, the error occurs. • If the index is zero, the above checking is canceled.
Identify signal	This signal used for tool identification. Available I/O type = DI / RI / GI • This signal is check before tool attach/detach instruction. If this signal is OFF, the error occurs. • If the index is zero, the above checking is canceled.
Battery	Select battery type. ON = Battery backup type OFF = Batteryless type NOTE To change multi-axis servo tool, Battery must be set to ON.

(Select the following when Battery is set to OFF in the common settings.)

Calibration Motion Type	On battery-less tools, the position (pulse value) is lost when the tool is detached. So, calibration must be carried out when tool is attached. Calibration involves the following two operations: • Turning the motor two turns for positioning the Pulsecoder (This operation is simply referred to as "positioning" from here on.) • Calibrating at a preset reference position The user selects from the following types, 1 to 6, to suit the user's particular system requirements. (See Table 32.4 and Fig. 32.4.)

Calibration motion type Selection Criteria

- Types 1, 2

Select these types on systems capable of holding the tool detachment position as a result of braking the servo tool axis motor up to the next servo tool attachment.

> **NOTE**
> With type 1 and 2, if calibration is interrupted by a hold or emergency stop, reference position data is lost. In such a case, see Subsection 32.13.2, "The Robot Stopped during Calibration".

When these types are set, the position where the tool is detached is automatically memorized as the reference position, and calibration is carried out at that position after the tool is attached. For this reason, the position when the tool is attached must be the same as position when it is detached.
The allowable errors of the detach and attach positions must fall within the respective following ranges for the direct-acting axis and rotary axis:

Direct-acting axis:
±(gear ratio/2) mm or less around the detach position
Rotary axis:
±((360/gear ratio)/2) deg or less around the detach position

Note that positional misalignment occurs during tool attachment if the above ranges are exceeded.

The difference between types 1 and 2 is in the operating direction during calibration. (See Table 32.4 and Fig. 32.4.)

Type 1:
Specify this type when the tool is detached at a position where it is impossible to move the tool in the positive direction of manual feed by two or more turns of the motor.
Type 2:
Specify this type when the tool is detached at a position where it is impossible to move the tool in the negative direction of manual feed by two or more turns of the motor.

- Types 3, 4

Select these types on systems not capable of holding the tool detachment position as a result of the servo tool axis motor not having a brake up to the next servo tool attachment. On such a system, the reference position must be set to another position as the tool detachment position cannot be used as the reference position.

With these types, contact of the tool axis with the open limit is detected by using the motor torque, and the contact position is set as the reference position. (This is because, in the standard setting, the calibration motion causes the tool axis to move in the negative direction of manual feed.) For details on how to set the reference position, see Section 32.5, "SETTING THE REFERENCE POSITION".

The difference between types 3 and 4 is in the operating direction during calibration. (See Table 32.4 and Fig. 32.4.)

Type 3:
Specify this type when the tool is detached at a position where the distance to the open limit is two or more turns of the motor.

Type 4:
Specify this type when the tool is detached at a position where the distance to the open limit is two or less turns of the motor.

NOTE

1 In the calibration, to move the tool axis in the positive direction of manual feed (to use the limit on the positive side of manual feed), change the following system variable to 0:
$SVTCTOOLxx.$CLOSDIR = 1 → 0 (xx: Tool number)

2 With types 3 and 4, it is necessary to expand the stroke limit so that the tool axis touches the open limit.

- Types 5, 6

Like types 3 and 4, specify these types on systems not capable of holding the tool detachment position as a result of the servo tool axis motor not having a brake up to the next servo tool attachment. With types 5 and 6, however, a detection signal must be provided on the tool by means of a limit switch, and the contact position must be set as the reference position. Install the detection switch in such a place that the detection switch is turned on when the tool axis is moved in the negative direction of manual feed (this is because, in the standard setting, the calibration motion causes the tool axis to move in the negative direction of manual feed). For details on how to set the reference position, see Section 32.5, "SETTING THE REFERENCE POSITION".
The difference between types 5 and 6 is in the operating direction during calibration. (See Table 32.4 and Fig. 32.4.)

Type 5:
Specify this type when the tool is detached at a position where the distance to the position where the detect signal turns ON is two or more turns of the motor.

Type 6: Specify this type when the tool is detached at a position where the distance to the position where the detect signal turns ON is two or less turns of the motor.

NOTE

1 In the calibration, to move the tool axis in the positive direction of manual feed, change the following system variable to 0:
$SVTCTOOLxx.$CLOSDIR = 1 → 0 (xx: Tool number)

2 With types 5 and 6, a switch such as a limit switch must be installed in the tool.

Table 32.4 Calibration motion type

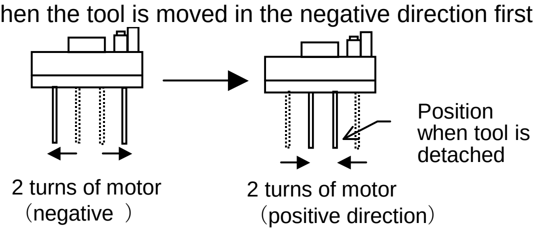
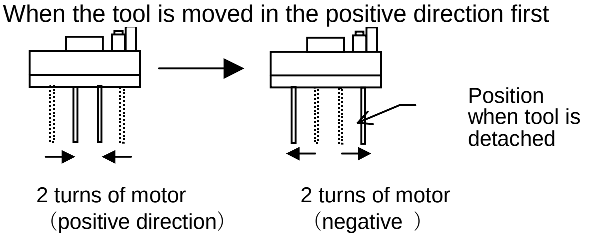
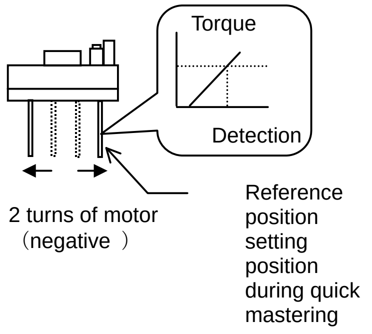
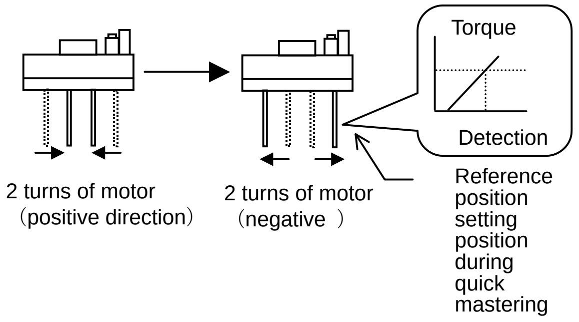
Type	Calibration Motion ($SVTCTOOLxx.$CLOSDIR is 1.)	Set Position of Reference position
Type 1	The tool is moved in the negative direction of manual feed by two turns of the motor and is subsequently moved in the positive direction by two turns, then the tool is calibrated. (NOTE 1) Set setting item (1) on the next page.	Position when tool is detached Reference position is not required. (NOTE 2)
Type 2	The tool is moved in the positive direction of manual feed by two turns of the motor and is subsequently moved in the negative direction by two turns, then the tool is calibrated. Set setting item (1) on the next page. (NOTE 1)	Position when tool is detached Reference position is not required. (NOTE 2)

Type	Calibration Motion ($SVTCTOOLxx.$CLOSDIR is 1.)	Set Position of Reference position
Type 3	The tool is moved in the negative direction of manual feed until touch torque is detected, then the tool is calibrated. Arrival at the limit is detected by motor torque. Set setting items (1), (2) and (3) on the next page.	Open limit position
Type 4	The tool is moved in the positive direction of manual feed by two turns of the motor and is subsequently moved to the limit on the negative side until touch torque is detected, then the tool is calibrated. Arrival at the limit is detected by motor torque. Set setting items (1), (2) and (3) on the next page.	Open limit position
Type 5	The tool is moved in the negative direction of manual feed until the detection signal is turned on, then the tool is calibrated. Set setting items (1), (2) and (4) on the next page.	Position where detection signal turns ON
Type 6	The tool is moved in the positive direction of manual feed by two turns of the motor and is subsequently moved in the negative direction until the detection signal is turned on, then the tool is calibrated. Set setting items (1), (2) and (4) on the next page.	Position where detection signal turns ON

NOTE

1 Tool master data is lost if the operation is canceled due to a hold or an emergency stop during calibration that is carried out during the tool change sequence when types 1 or 2 are selected. If this happens, manually carry out zero position mastering.

2 With types 1 and 2, the system automatically memorizes the position where the servo tool is detached as the reference position. This memorized reference position is used in calibration when the same tool is next attached. For this reason, when types 1 or 2 are selected, the detachment position and the attachment position must be the same. Do not select types 1 or 2 when a motor without a brake is used for the servo tool axis.

Type 1	Type 2
When the tool is moved in the negative direction first  Position when tool is detached 2 turns of motor (negative) 2 turns of motor (positive direction)	When the tool is moved in the positive direction first  Position when tool is detached 2 turns of motor (positive direction) 2 turns of motor (negative)
Type 3	**Type 4**
When the tool is moved in the negative direction only  Torque Detection 2 turns of motor (negative) Reference position setting position during quick mastering	When the tool is moved in the positive direction then moved in the negative direction  Torque Detection 2 turns of motor (positive direction) 2 turns of motor (negative) Reference position setting position during quick mastering

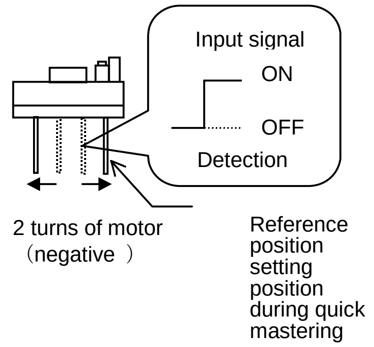
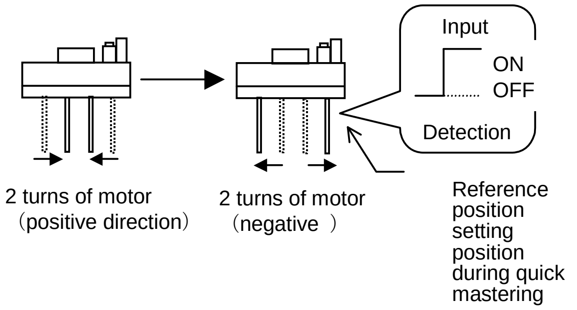
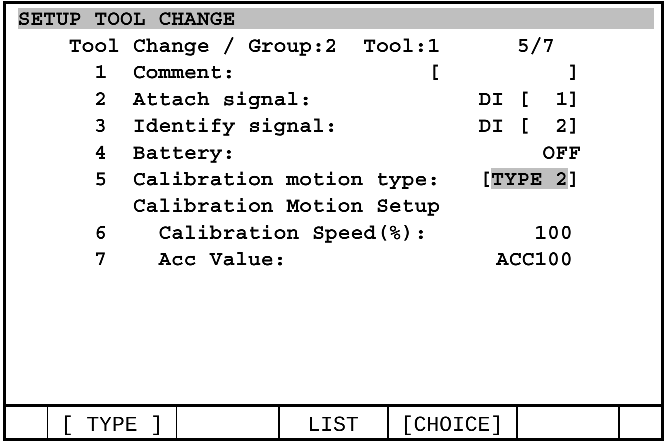
Type 5	Type 6
When the tool is moved in the negative direction only  Input signal ON OFF Detection 2 turns of motor （negative ） Reference position setting position during quick mastering	When the tool is moved in the positive direction then moved in the negative direction  Input ON OFF Detection 2 turns of motor （positive direction） 2 turns of motor （negative ） Reference position setting position during quick mastering

Fig. 32.4 Calibration motion type

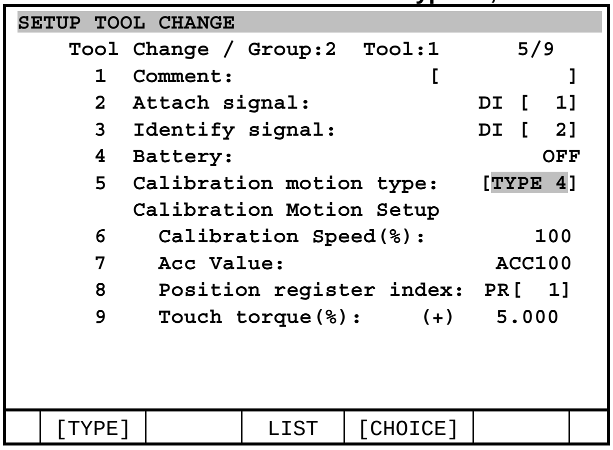
The positive and negative directions in the above figure are determined on the assumption that $SVTCTOOLxx.$CLOSDIR = 1 is set. If this system variable is set to 0, the directions are reversed.

Screen for calibration types 1, 2



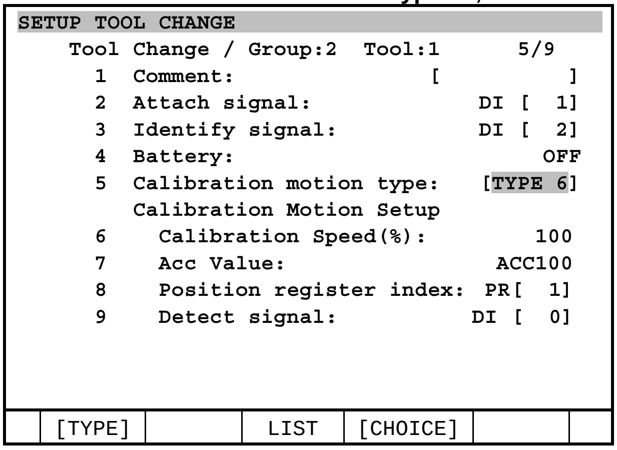
```
SETUP TOOL CHANGE
    Tool Change / Group:2  Tool:1      5/7
      1  Comment:               [          ]
      2  Attach signal:             DI [  1]
      3  Identify signal:           DI [  2]
      4  Battery:                        OFF
      5  Calibration motion type:   [TYPE 2]
         Calibration Motion Setup
      6    Calibration Speed(%):         100
      7    Acc Value:                 ACC100

   [ TYPE ]          LIST    [CHOICE]
```

Screen for calibration types 3, 4



```
SETUP TOOL CHANGE
    Tool Change / Group:2  Tool:1      5/9
      1  Comment:               [          ]
      2  Attach signal:             DI [  1]
      3  Identify signal:           DI [  2]
      4  Battery:                        OFF
      5  Calibration motion type:   [TYPE 4]
         Calibration Motion Setup
      6    Calibration Speed(%):         100
      7    Acc Value:                 ACC100
      8    Position register index:  PR[  1]
      9    Touch torque(%):     (+)    5.000

  [TYPE]          LIST   [CHOICE]
```

Screen for calibration types 5, 6



```
SETUP TOOL CHANGE
    Tool Change / Group:2  Tool:1      5/9
      1  Comment:               [          ]
      2  Attach signal:             DI [  1]
      3  Identify signal:           DI [  2]
      4  Battery:                        OFF
      5  Calibration motion type:   [TYPE 6]
         Calibration Motion Setup
      6    Calibration Speed(%):         100
      7    Acc Value:                 ACC100
      8    Position register index:  PR[  1]
      9    Detect signal:           DI [  0]

  [TYPE]          LIST   [CHOICE]
```

(1) Open limit operation setting (Set on all calibration types.)

Calibration speed	Specify the calibration motion operation speed.
Accelerate rate	Specify the calibration motion accelerate rate. Default is 100(%).

(2) Position register No. setting
(Set as follows when types 3 to 6 are set in the Calibration Setup.)

Position register No	Position register for holding the momentary position where arrival at the reference position (open limit or signal input position) was detected Set the position register specified here exclusively for calibration. (Do not use the same position register in the program, for example.)

(3) Open limit detection torque setting
(Set as follows when types 3 or 4 are set in the Calibration Setup.)

Touch torque (%)	Threshold value of the motor torque for detecting arrival at the open limit Set the value within range 0.0 to 100.0%.

(4) Detection signal setting (Set as follows when types 5 or 6 are set in the Calibration Setup.)

Detection signal	Type and No. of input signal for detecting arrival at the reference position • DI • RI Specify a No. available for each of the signals as the signal No.

32.5 SETTING THE REFERENCE POSITION

32.5.1 Battery-less Type Tools

On battery-less tools, the position (pulse value) is lost when the tool is detached. So, calibration must be carried out when tool is attached.
The position taken as the reference for calibration is called the "reference position".
On battery-less tools, the reference point must be set before the tool is detached.
How the reference point is set differs according to the calibration motion type.

Types 1, 2

The system automatically sets the reference position when the servo tool is detached.

Types 3, 4

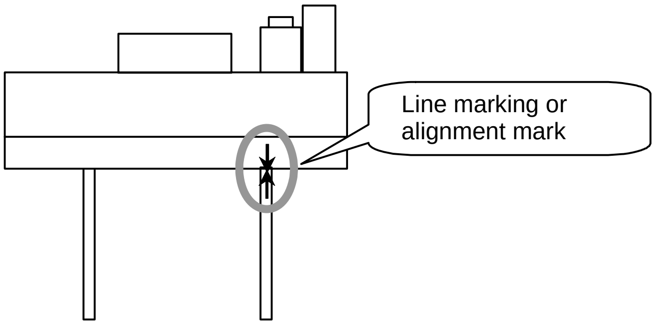
Create a program for detecting the reference position using the touch torque, and execute this program to calculate the reference position. For details, refer to Subsection 32.12.1.

Types 5, 6

Create a program for detecting the reference position using the detection signal, and execute this program to calculate the reference position. For details, refer to Subsection 32.12.2.

32.5.2 Battery-Mounted Type Tools

On battery-mounted type tools, the position (pulse value) is not lost even if the tool is detached. Therefore, the reference position used for calibration when the tool is attached need not be set as it needs to be on a battery-less type tool. However, that we recommend setting a reference position for quick mastering by marking a line or alignment mark as shown in the figure below for extra safety in the event that the battery comes loose. For details on how to set the reference position for quick mastering, refer to 32.12.3.



32.6 TOOL CHANGE INSTRUCTION

For changing servo tools, the following instructions are provided.

32.6.1 TOOL DETACH Instruction

This instruction can detach the servo tool without power off.

Format

TOOL DETACH[i]

i : Tool number (1 to 4)
Specify the tool number which you want to detach.
R[ ] and AR[ ] can be used.
Ex) TOOL DETACH[1]
TOOL DETACH[R[2]]
TOOL DETACH[AR[1]]

32.6.2 TOOL ATTACH Instruction

This instruction can attach the servo tool without power off. In case of batteryless tool, the mastering and calibration are performed automatically.

Format

TOOL ATTACH[i]

i : Tool number (1 to 4)
Specify the tool number which you want to detach.
R[ ] and AR[ ] can be used.
Ex) TOOL ATTACH[4]
TOOL ATTACH[R[3]]
TOOL ATTACH[AR[1]]

32.6.3 Sample Program

Sample program

	[Explanation]
1: J P[1:Approach] 100% FINE	1: Move to approach position.
2: L P[2:TOOL1 SET] 100mm/sec FINE	2: Set tool axis to 0 position.
3: TOOL DETACH[1]	3: Detach tool 1.
4: L P[1:Approach] 100mm/sec FINE	4: Release tool1.
:	
11: J P[11: Approach] 100% FINE	
12: L P[12:TOOL2 SET] 100mm/sec FINE	

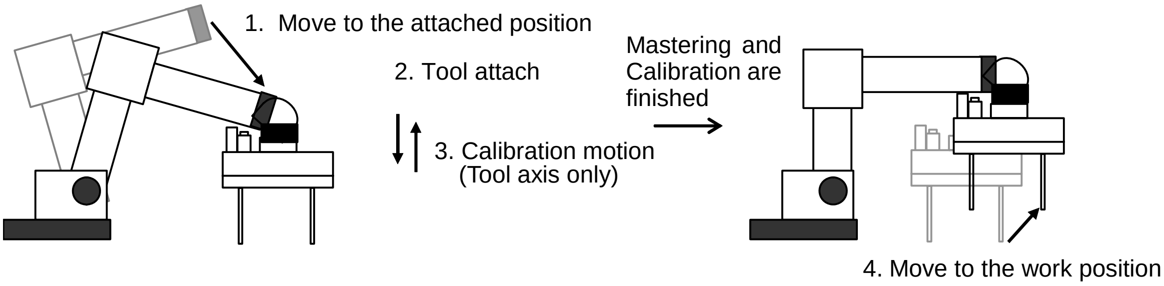
13: TOOL ATTACH[2] 14: UTOOL_NUM=2 15: L P[13: Approach] 100mm/sec FINE : 99: END	13: Attach tool 2. 14:Change tool frame number. 99: Normal end

NOTE

The above is merely a sample program. Create programs to suit the requirements of your system.

32.6.4 Forward Execution

It is necessary to check that the tool number of instruction is same as the actual tool. And in the case of attach instruction, the calibration motion type must be set correct type.
In the case of batteryless tool, the calibration motion (3 in the following figure) is done.



NOTE

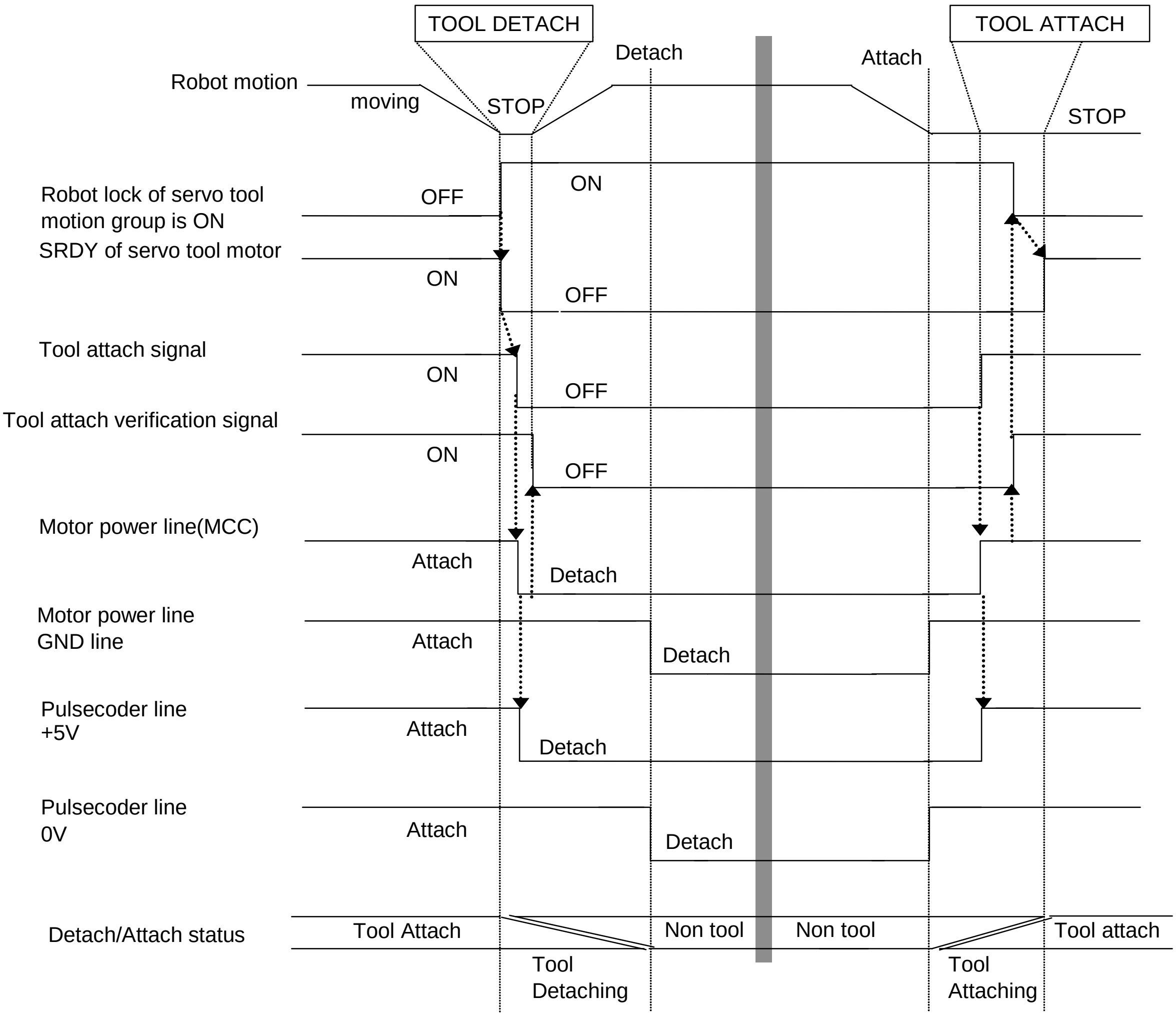
In the case of calibration motion type is Type 1 or Type 2, if the motion is stopped by e-stop or HOLD during the calibration motion, the mastering data is lost. If the mastering data is lost, please do the zero mastering and calibration by manual again.
Otherwise,
1) The current position on TP is not matched to the actual position.
2) If the motion instruction is executed, the following alarm is occurred.
"MOTN-049 Attempt to move w/o calibrated"
3) The TOOL DETACH instruction does not work until manual mastering and calibration are finished.

In the case of TOOL DETACH instruction, both robot and servo tool do not move at all.

32.6.5 Backward Execution

Both TOOL ATTACH and TOOL DETACH instruction can not be execute by backward execution.

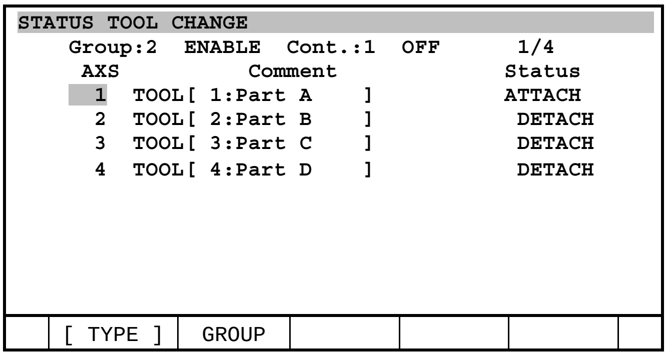
32.7 TOOL CHANGE SEQUENCE



32.8 TOOL CHANGE STATUS

On the tool change status screen, verify that which tool is attached or detached now.

1 Press [MENU] key.
2 Select “0 next page”, ”4 STATUS”.
3 Press F1, [TYPE].
4 Select “Tool change”.



```
STATUS TOOL CHANGE
    Group:2  ENABLE  Cont.:1  OFF      1/4
     AXS          Comment             Status
       1  TOOL[ 1:Part A    ]         ATTACH
       2  TOOL[ 2:Part B    ]          DETACH
       3  TOOL[ 3:Part C    ]          DETACH
       4  TOOL[ 4:Part D    ]          DETACH
```

The following status are displayed.

AXS	The axis number of the motion group which is assigned to the current motion group.
Tool	The tool number which is assigned to the above axis.
Status	Tool status ATTACH : The tool is attached. DETACH : The tool is detached.
Comment	The comment of each tool.

32.9 TEACHING

In the tool change system, please note the following points.

32.9.1 Notice for Teaching

Position data

In the tool change system, the each tool is assigned to the each axis of the servo tool motion group. For example, if three kinds of tool are assigned to tool change system, the servo tool motion group has three axes data. This means three axes data are recorded when servo tool position is thought.

(Ex. The position data of tool1 which is assigned to first axis)

```
GP1:     UF : 0, UT : 1, Config : 'N, 0, 0, 0',
  X =   1879.25 mm,   Y = -1035.37 mm,   Z =     162.65 mm,
  W =   -178.67 deg,   P =   -13.68 deg,   R =     -37.32 deg
GP2:     UF : 0, UT : 1,
   J1 = 10.00 mm      J2 = 0.00 mm     J3 = 0.00 mm
```

Please note that this position data has not only tool 1 data but also tool 2 and tool 3 data. This means this position data can move tool 2 and tool 3. If tool 2 is attached, tool 2 moves to 0mm position. This specification is for the flexibility of the program. But, if the user attaches different tool, it may cause interfere of tool. To prevent this problem, please specify different tool number to the tool close direction for each tool.

By this function, the position data for each tool has the different tool number. Because of this, even if the user executes the program with different tool, the program is stopped by “INTP-253 Tool frame number mismatch”.

Re-teaching after the servo tool is added

When the servo tool is added, the program made before adding cannot use. Please re-teach positions on attaching the corresponding tool.

32.9.2 Sample Program

Sample program

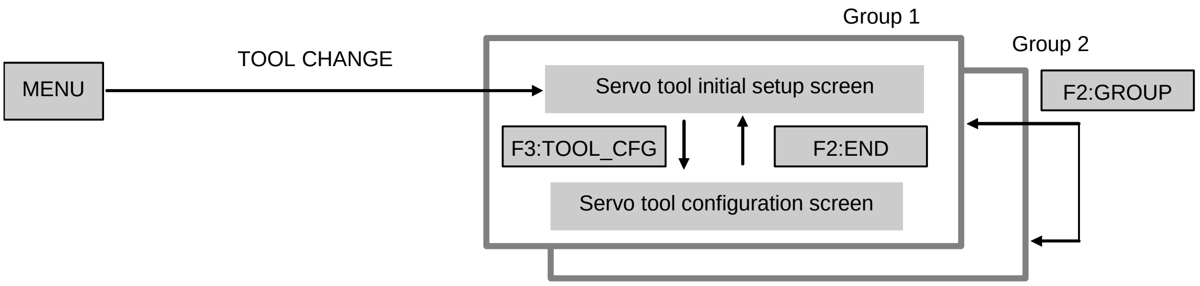
:	[Explanation]	
2: **TOOL ATTACH[1]**	2:Attach tool1.	
3: **UTOOL_NUM=1**	3:**Change tool frame number to 1.**	Move with tool number 1.
:	:	
11:J P[11] 100% FINE	11:Handle with tool1.	
12:L P[12] 100mm/sec FINE	12:Handle with tool1.	
13:J P[13:Approach] 100% FINE	13:Move to approach position.	
14:L P[14:Tool1 set] 100mm/sec FINE	14:Set tool1 to tool stocker.	
15: **TOOL DETACH[1]**	15:Detach tool1.	
16:L P[13:Approach] 100mm/sec FINE	16:Release tool1.	
:	:	
21:J P[21:Approach] 100% FINE		
22:L P[22:Tool2] 100mm/sec FINE		
23: **TOOL ATTACH[2]**	23:Attach tool2.	
24: UTOOL_NUM=2	24:Change tool frame number to 2.	Move with tool number 2.
25:L P[23:Approach] 100mm/sec FINE		
26:J P[24] 100% FINE	26:Handle with tool2.	
27:L P[25] 100mm/sec FINE	27:Handle with tool2.	
:	:	
99: END	99:Normal end	

NOTE

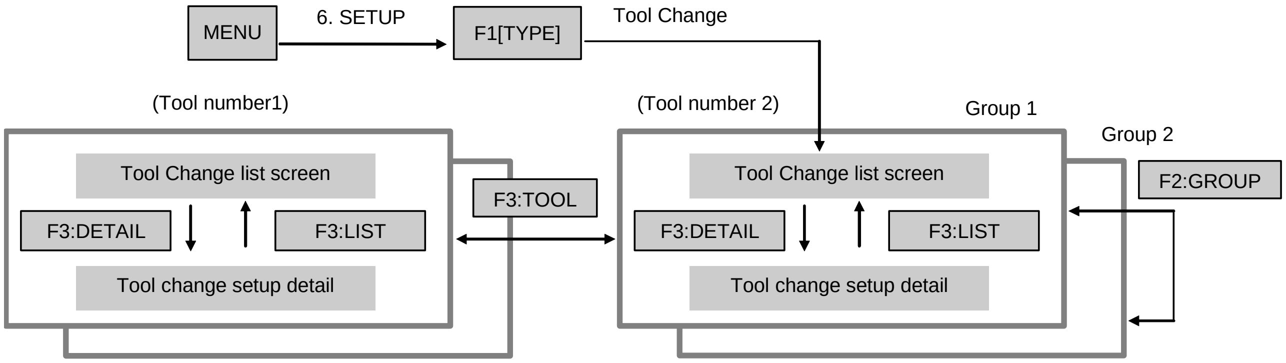
This is sample program. Create the program related to your system.

32.10 CONSTRUCTION OF SERVO TOOL CHANGE SCREEN

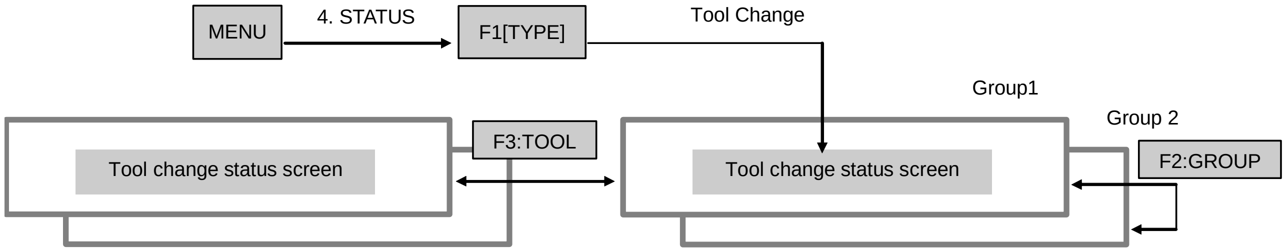
INITIAL SETUP SCREEN（This screen can be displayed at CONTROLLED START Menu.）


Group 1
Group 2
MENU
TOOL CHANGE
Servo tool initial setup screen
F2:GROUP
F3:TOOL_CFG
F2:END
Servo tool configuration screen

SETUP SCREEN



STATUS SCREEN



32.11 TOOL CHANGE INITIAL SETUP

The following explains how to make initial settings required for tool change system installation.

> **NOTE**
> The setup procedure explained below is required just once when the system is installed. Once settings are made, they need not be changed in ordinary use.

32.11.1 Setting Motion Parameters for Servo Tool Axes

Set tool axes used for the tool change function within each operation group according to the procedure below.

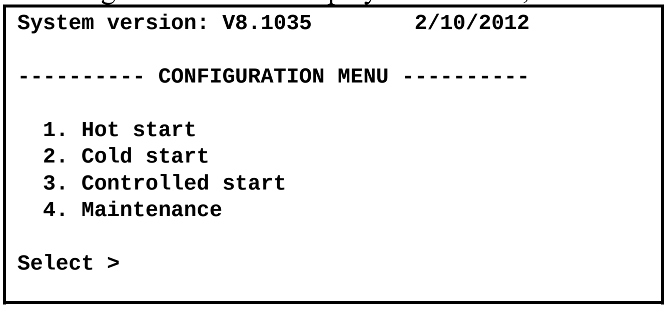
- 1st to 4th axes: Independent axis setting procedure

32.11.2 Assigning Tool Numbers to Servo Tool Axes

Open the servo tool initial setup screen from the controlled start menu, and set the following:

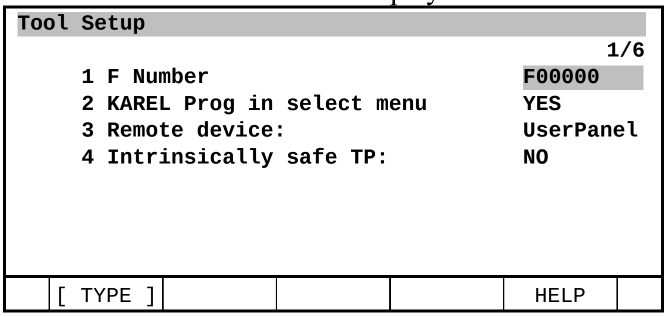
1 Turn off the power. While holding [PREV] key and [NEXT] key, turn on the power.

2 After a while, the following screen will be displayed. Select 3, Controlled start.



```
System version: V8.1035        2/10/2012

---------- CONFIGURATION MENU ----------

  1. Hot start
  2. Cold start
  3. Controlled start
  4. Maintenance

Select >
```

3 After a while, the controlled start screen will be displayed as follows.

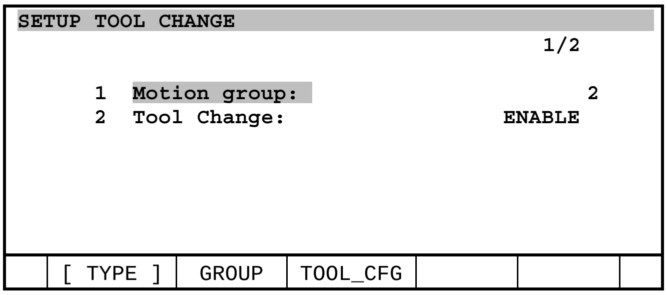


```
Tool Setup
                                          1/6
     1 F Number                       F00000
     2 KAREL Prog in select menu      YES
     3 Remote device:                 UserPanel
     4 Intrinsically safe TP:         NO
```

4 Press [MENU] key to display the screen menu.

5 Select 0, NEXT then 4, TOOL CHANGE. The servo tool initial setup screen on the next page will be displayed .

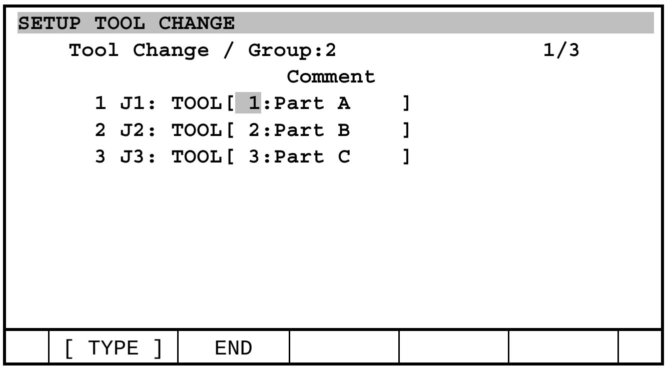
Motion Group	Input the operation group number with which servo tools are set.
Tool Change	Enable or disable the tool change function.



```
SETUP TOOL CHANGE
                                                    1/2

      1  Motion group:                                2
      2  Tool Change:                           ENABLE







   [ TYPE ]   GROUP   TOOL_CFG
```

The tool change function switches servo tools that are set within the same operation group. Note that it is impossible to switch between tools that are set in different operation groups.

6 Press F3, TOOL_CFG, then the servo tool configuration screen will be displayed.



```
SETUP TOOL CHANGE
    Tool Change / Group:2                           1/3
                     Comment
      1 J1: TOOL[ 1:Part A    ]
      2 J2: TOOL[ 2:Part B    ]
      3 J3: TOOL[ 3:Part C    ]






   [ TYPE ]    END
```

Set the items listed below for the tool axes included in the selected operation group.
If there is no problem with using the standard settings, the standard settings need not be modified.

TOOL	Set a number from 1 to 4.
Comment	Set a comment on the tool number. You can also set a comment on the tool change setup screen after cold start.

To change multi-axis servo tool, assign the tool number to multiple axes.

NOTE

- The tool number must be assigned to successive axes for multi-axis servo tool change.
- Four (or more)-axis servo tool cannot be changed.
- The servo tool cannot be changed to the one with a different number of axes.

32.11.3 Setting System Variables

To use servo guns on a system with the servo tool change function installed, set the following system variables to TRUE.

1. $SGCFG[equipment number of servo gun].$DUAL_AXIS[1] = FALSE -> TRUE

2. $SGCFG[1].$DUAL_AXIS[2] = FALSE -> TRUE

> **NOTE**
> The setting 1 must be done for all servo gun equipment.

32.12 TOOL CHANGE REFERENCE POSITION SETUP METHOD (BATTERY-LESS TYPE)

32.12.1 Reference Position Setup for Calibration Types 3 and 4

Reference Position Setup Program

Create the following program to calculate the reference position based upon the sign of the calculated touch torque.

> **NOTE**
> Only the servo tool axis operation group can be used as the operation group of the program to be used for setting the reference position.
> Example) When the servo tool axis operation group is the 2nd group:
> The operation group becomes [*,1,*,*,*,*,*,*].

Sample Program Operation Group=[*,1,*,*,*,*,*,*]

	[Explanation]
1: J P[1:ZERO POS] 10% FINE	1: Servo tool axis 0 position (The position need not necessarily be the 0 position.)
(when sign of touch torque is plus) 2: SKIP CONDITION $MISC[g].$HPD_TRQ[ga] > 2 *1) (when sign of touch torque is minus) 2: SKIP CONDITION $MISC[g].$HPD_TRQ[ga] < -2 *1)	2: As the touch torque in the skip condition, set the threshold value of the torque from which arrival at the limit can be detected when the tool moved in the negative direction of manual feed touches the limit. Pay attention to the sign of the touch torque.
3: J P[2:TOOL OPEN END] 1% *2) FINE ACC 100 *3)SKIP, LBL[1],PR[1]=JPOS	3: Move the tool in the negative direction of manual feed to touch the limit. If detection fails, jump to LBL[1].
4: END	4: Normal end
5: LBL[1]	5: Jump to here if torque cannot be detected.
6: UALM[1]	6: Generate user alarm.

> **NOTE**
> The above is merely a sample program. Create programs to suit the requirements of your system.
> g: Operation group number of currently attached tool
> ga: Axis number of currently attached tool
> If the currently attached servo tool axis is taken as the 1st axis of the 2nd group, then g and ga are as follows: g=2, ga=1

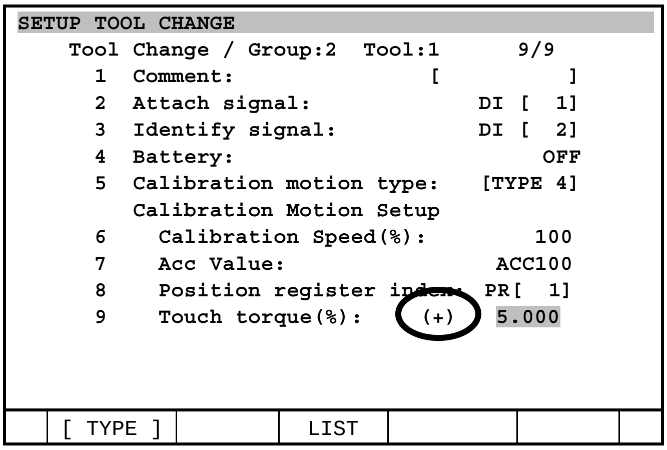
Program Operation Adjustment Method

Adjust the value at the shaded section in the sample program to adjust program operation. Operation adjusted here is used as calibration motion that is carried out when attaching the tool.

*1) Touch torque
Increase the value from about ±2, and calculate the value at which erroneous detection of contact does not occur.

The sign of touch torque is the sign that is currently displayed at Touch torque in the following Calibration Setup screen.

> **NOTE**
> After confirming the sign in this screen, teach it to the program. Do not make a mistake when entering the sign. If you do so, contact with the open limit may not be detected.



```
SETUP TOOL CHANGE
    Tool Change / Group:2  Tool:1      9/9
      1  Comment:              [          ]
      2  Attach signal:            DI [  1]
      3  Identify signal:          DI [  2]
      4  Battery:                       OFF
      5  Calibration motion type:   [TYPE 4]
         Calibration Motion Setup
      6    Calibration Speed(%):        100
      7    Acc Value:                ACC100
      8    Position register index:  PR[  1]
      9    Touch torque(%):     (+)   5.000
```

*2) Program operation speed
To set the same calibration motion that is carried out when attaching the tool, execute this program at the same override (normally, 100%) as when the attach instruction is executed for the tool used during actual production. For this reason, reduce the program operation speed so that contact is made with the open limit at a safe speed even if the program is executed at a high override.
For added safety, increase the program operation speed value successively from 1%.

*3) Accelerate rate
Decrease the accelerate rate if motion is quick even if the program operation speed is 1%.

Reference Position Setup

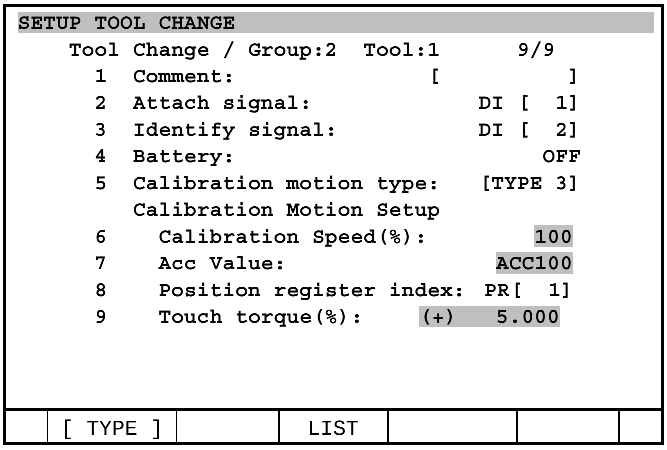
After you have adjusted operation in the program, execute the program to bring the servo tool axis into contact with the open limit.

Set up the reference position according to the procedure described in 32.12.3 with the servo tool contacting the open limit.

Tool Change Setup

To set operation in the newly adjusted program to the same motion as calibration motion that is carried out when attaching the tool, reflect the program operation speed, accelerate rate and touch torque adjustment values in the tool change setup.

1 Press [MENU] key to display the screen menu.
2 Select 6 SETUP.
3 Press F1, [TYPE] to display the screen selection menu.
4 Select Tool Change. The Tool Change Setup List screen will be displayed.
5 Select the tool to be adjusted from this list, and press F3, DETAIL to display the Tool Change Details Setup screen.
6 Set the operation speed, accelerate rate and touch torque adjustment values at the shaded sections in the following screen:



```
SETUP TOOL CHANGE
    Tool Change / Group:2  Tool:1       9/9
      1  Comment:               [          ]
      2  Attach signal:             DI [  1]
      3  Identify signal:           DI [  2]
      4  Battery:                        OFF
      5  Calibration motion type:   [TYPE 3]
         Calibration Motion Setup
      6    Calibration Speed(%):         100
      7    Acc Value:                 ACC100
      8    Position register index:  PR[  1]
      9    Touch torque(%):     (+)   5.000

   [ TYPE ]          LIST
```

32.12.2 Reference Position Setup for Calibration Types 5 and 6

Provision of Detection Signal

With calibration types 5 and 6, the position where the detection signal is input is taken as the reference position during quick mastering.

The user must install a limit switch for providing the detection signal capable of assuring the absolute position of the tool.

Reference Position Setup Program

Create the following program to calculate the reference position based upon the provisioned detect signal.

> **NOTE**
>
> Only the servo tool axis operation group can be used as the operation group of the program to be used for setting the reference position.
> Example)
> When the servo tool axis operation group is the 2nd group: The operation group becomes [*,1,*,*,*,*,*,*].

Sample Program Operation Group=[*,1,*,*,*,*,*,*]

1:J P[1:ZERO POS] 10% FINE 2:SKIP CONDITION DI[1*1)] =ON 3:J P[2:SWITCH POS] 1% *2) FINE ACC100 *3)SKIP, LBL[1],PR[1]=JPOS 4:END 5:LBL[1] 6:UALM[1]	[Explanation] 1: Servo tool axis 0 position (The position need not necessarily be the 0 position.) 2: Set turning ON of the provided detection signal as the scan condition. 3: Move tool to position where detection signal turns ON. (Jump to label[1] if the detection signal does not turn ON.) 4: Normal end 5: Jump to here if detection signal does not turn ON. 6: Generate user alarm

> **NOTE**
>
> The above is merely a sample program. Create programs to suit the requirements of your system.

Program Operation Adjustment Method

Adjust the value at the shaded section in the sample program to adjust program operation.
Operation adjusted here is used as calibration motion that is carried out when attaching the tool.

*1) Detection Signal
Assign the provided detection signal to DI[ ] or RI[ ].

*2) Program Operation Speed

To set the same calibration motion that is carried out when attaching the tool, execute this program at the same override (normally, 100%) as when the attach instruction is executed for the tool used during actual production. For this reason, reduce the program operation speed so that the switch will not be broken even if the program is executed at a high override.

For added safety, increase the program operation speed value successively from 1%.

*3) Accelerate rate

Decrease the accelerate rate if motion is quick even if the program operation speed is 1%.

Reference Position Setup

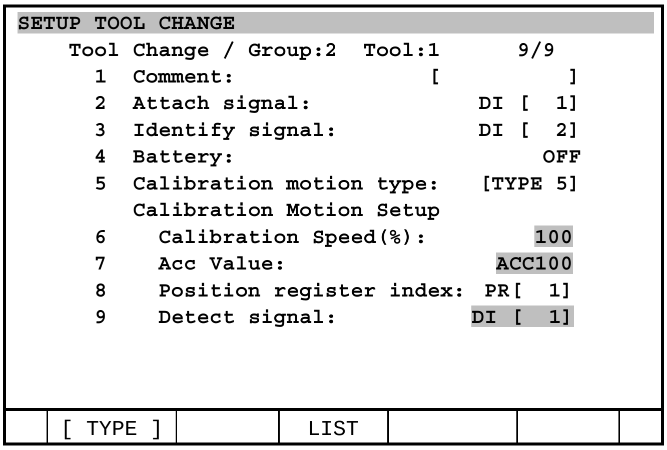
After you have adjusted operation in the program, execute the program to stop the servo tool axis at the position where the detection signal turns ON.

Set up the reference position according to the procedure described in 32.12.3 as it is.

Tool Change Setup

To set operation in the newly adjusted program to the same motion as calibration motion that is carried out when attaching the tool, reflect the program operation speed, accelerate rate and detection signal adjustment values in the tool change setup.

1 Press [MENU] key to display the screen menu.
2 Select 6 SETUP.
3 Press F1, [TYPE] to display the screen selection menu.
4 Select Tool Change. The Tool Change Setup List screen will be displayed.
5 Select the tool to be adjusted from this list, and press F3, DETAIL to display the Tool Change Details Setup screen.
6 Set the operation speed, accelerate rate and detection signal adjustment values at the shaded sections in the following screen:



```
SETUP TOOL CHANGE
    Tool Change / Group:2  Tool:1      9/9
      1  Comment:              [          ]
      2  Attach signal:            DI [  1]
      3  Identify signal:          DI [  2]
      4  Battery:                       OFF
      5  Calibration motion type:   [TYPE 5]
         Calibration Motion Setup
      6    Calibration Speed(%):        100
      7    Acc Value:                ACC100
      8    Position register index:  PR[  1]
      9    Detect signal:           DI [  1]
```

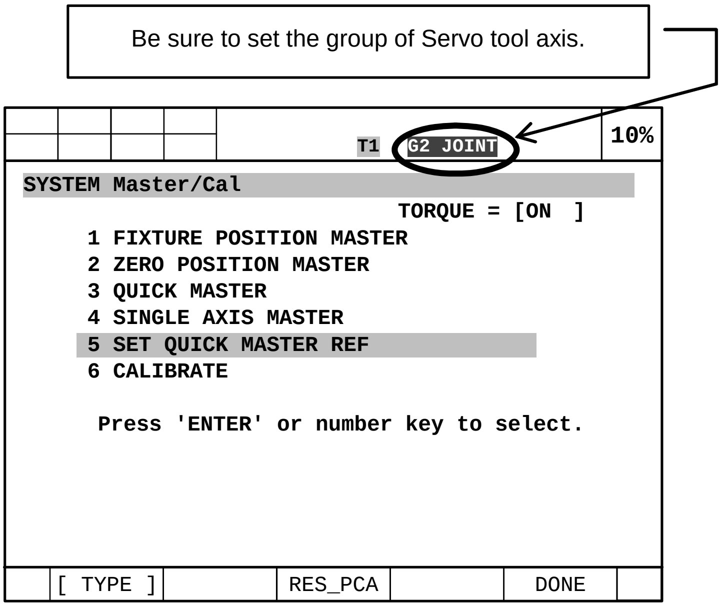
32.12.3 Quick Mastering Reference Position Setup

Calibration (quick mastering) is automatically carried out when the servo tool is changed. However, the reference points must be set to each of the tools to enable automatic calibration. The reference points are set with the robot in a cold start state.

Follow the procedure below to set the reference points:

1 Press [FCTN] key to select 3 CHANGE GROUP. Change the operation group to the desired servo tool group (e.g. 2nd group).
2 Press [MENU] key to display the screen menu.
3 Press 0 NEXT and 6 SYSTEM.
4 Press F1, [TYPE] to display the Screen Selection menu.
5 If Master/Cal is not displayed in the menu, select Variables to display the system variables screen. Set the $MASTER_ENB system variable to "1".

6 Select Master/Cal at 4. to display the calibration screen.
7 Select "5 SET QUICK MASTER REF" and F4, YES.
8 The message "Quick master reference point is set." is displayed. This indicates that setting of the reference point is completed.


Be sure to set the group of Servo tool axis.

```
                                  T1  G2 JOINT              10%
SYSTEM Master/Cal
                                  TORQUE = [ON  ]
     1 FIXTURE POSITION MASTER
     2 ZERO POSITION MASTER
     3 QUICK MASTER
     4 SINGLE AXIS MASTER
     5 SET QUICK MASTER REF
     6 CALIBRATE

      Press 'ENTER' or number key to select.

 [ TYPE ]          RES_PCA            DONE
```

32.13 TROUBLESHOOTING

32.13.1 The Attach Instruction is Executed when the Tool is not Attached.

Symptom

The tool is not actually attached. So, a communications alarm such as "SRVO-068 DTERR alarm" occurs on the Pulsecoder. Correct this by Remedy 1 or Remedy 2 below according to the present conditions.

Remedy 1

1 Turn the machine lock ON in the servo tool operation group (normally, group 2) in the Test Cycle screen.
2 Move the robot by manual feed so that the robot and tool are physically attached.
3 Cycle power of the controller.

Battery-less tool (calibration motion types 1, 2)
4 The tool pulse values and mastering data are lost. So, first move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.
5 Then, carry out tool zero position mastering and calibration manually.

Battery-less tool (calibration motion types 3, 4, 5, 6)
6 The tool pulse values and mastering data are lost. So, move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.
7 Carry out tool zero position mastering and calibration manually. The tool 0 position may be an approximate position.
8 Execute the detach instruction to temporarily detach the tool. Also, physically detach the tool from the robot. Otherwise, the tool will not be calibrated at the next attach operation.
9 Execute the attach instruction again to attach the tool. Calibration this time is carried out at the reference position to restore the position.

Battery mounted tool

10 The tool is attached and positioned, so it can be used as it is.

Remedy 2

1 Directly change the system variable to detach the tool. To do this, set the following system variable to "0":
$SCR_GRP[servo tool operation group No. (normally, 2)].$AXISORDER[all 1 to 9] = 0

2 Cycle power of the controller.

3 The robot starts up with the tool in a detached state. Move the robot by manual feed to physically attach the robot and tool.

4 Execute the attach instruction to attach the tool.

5 The tool can be used as it is after it is attached regardless of the battery type.

32.13.2 The Robot Stopped during Calibration.

Symptom

This symptom occurs only on battery-less tools. Note that the symptom and remedy vary according to the preset calibration type.

Battery-less tool (calibration motion types 1, 2)
The "TOOL-012 Tool mastering data is lost" alarm has occurred.

Battery-less tool (calibration motion types 3, 4, 5, 6)
Alarm has not occurred.

Remedy

Calibration motion types 1, 2

1 The tool pulse values and mastering data are lost. So, first move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.
This alarm may not be occurring. If so, skip this step and proceed to step 2.

2 Then, carry out tool zero position mastering and calibration manually.

Calibration motion types 3, 4, 5, 6

1 Tool operation can be resumed as it is.

32.13.3 Calibration Motion Failed.

Symptom

The "TOOL-020 Calibrate motion is failed" alarm occurred.
This symptom occurs only when the calibration motion type is 3, 4, 5 or 6 on a battery-less tool.
This alarm occurs when the touch torque or detection signal conditions are not satisfied during calibration.
In both of the following Remedies 1 and 2, the tool is calibrated again. So, calibration motion will fail again if the touch torque or detection signal conditions are not satisfied. Before you remedy this symptom, make sure that the touch torque or detection signal conditions are correct.

Remedy 1

Carry out steps 6 to 9 of Subsection 32.13.1 Remedy 1.

Remedy 2

Same as Remedy 2 of Subsection 32.13.1.

32.13.4 A Different Tool from that Specified by the Attach Instruction is Attached.

Symptom

The tool is attached to the robot, but the tool's current position is the wrong value. On a battery-less tool, an alarm may cause the robot to stop during calibration.

Remedy

The following describes an example where tool 2 (actually attached tool) is attached to tool 1 (tool No. taught by the attach instruction).

1 In this state, the tool cannot be detached from the robot by the detach instruction. So, directly change the system variable to detach the tool. To do this, set the following system variable to "0":

$SCR_GRP[servo tool operation group No. (normally, 2)].$AXISORDER[all 1 to 9] = 0

2 Cycle power of the controller.
3 The tool starts up in a detached state.
4 Execute the TOOL ATTACH[2] instruction to attach tool 2. (In this state, tool 2 should be physically connected to the robot.)

When tool 2 is a battery-less tool and the calibration motion type is 1 or 2:
If tool 2 is moved when attached by mistake instead of tool 1, tool 2 will move and be at a position different to when it was detached. So, tool 2 may be calibrated to the wrong position by the attach instruction executed in step 4 above, resulting is misalignment between its actual and desired positions. In this case, carry out tool zero position mastering and calibration manually.

When tool 2 is other than the above:
Tool 2 can be used as it is.

5 Tool 1 can be used as it is by executing the attach instruction to attach the tool.

32.13.5 The Attached Tool has been Detached by Mistake (without Using the Detach Instruction).

Symptom

The tool is not actually attached. So, a communications alarm such as "SRVO-068 DTERR alarm" occurs on the Pulsecoder. Correct this by Remedy 1 or Remedy 2 below according to the present conditions.

Remedy 1

Same as Remedy 1 of Subsection 32.13.1

Remedy 2

Same as Remedy 2 of Subsection 32.13.1

32.13.6 The Tool Axis of a Detached Tool has Moved.

Symptom

Battery-less tool (calibration motion types 1, 2):
Misalignment has occurred. Misalignment occurs as a result of calibration at the wrong position when the tool is attached with the tool axis having moved to a different position. Note, however, that misalignment does not occur when tool axis movement is within the allowable errors indicated below.

The allowable error must fall within the respective following ranges for the direct-acting axis and rotary axis:

Direct-acting axis: ±(gear ratio/2)mm or less around the detach position
Rotary axis: ±((360/gear ratio)/2)deg or less around the detach position

Other instances:
No problem has occurred.

Remedy

Battery-less tool (calibration motion types 1, 2):

1 Execute the attach instruction to attach the tool.
2 If misalignment occurs, carry out tool zero position mastering and calibration manually.

32.13.7 The Battery Voltage has Fallen.

Symptom

The "SRVO-065 BLAL" alarm has occurred with the tool in an attached state.

Remedy

1 Change the battery with the tool attached.
2 The tool can be used as it is.

32.13.8 The Battery Ran Low while the Tool was Detached.

Symptom

The "SRVO-062 BZAL" and "SRVO-075 Pulse not established" alarms occurred when the tool was attached.

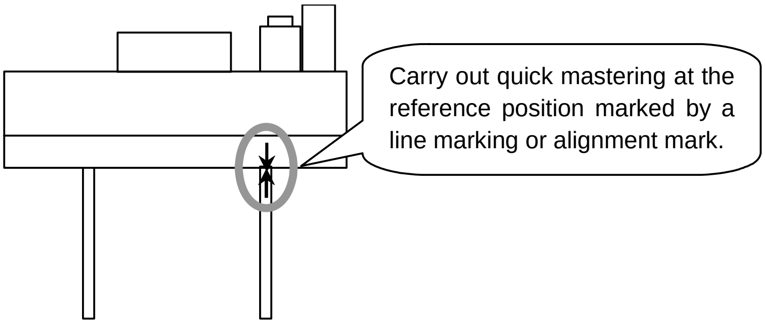
Remedy

When the reference position has not been set:

1 Press F3:RES_PCA in the Calibration screen, and select YES.
2 Cycle power of the controller.
3 Move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.
4 Carry out tool zero position mastering and calibration manually.

When the reference position has been set:

1 Press F3:RES_PCA in the Calibration screen, and select YES.
2 Cycle power of the controller.
3 Move the tool by manual feed to cancel the "SRVO-075 Pulse not established" alarm.
4 Move the tool by manual feed to the reference position marked by a line marking or alignment mark.
5 Execute “3 QUICK MASTER” in the Calibration screen.



Carry out quick mastering at the reference position marked by a line marking or alignment mark.

33 OPERATION WITHOUT SHIFT FUNCTION

This function makes it possible to operate without [SHIFT] key in case operation robot by Teach Pendant. Robot operation without shift function (J591) makes it possible to operate both jog operation and test execution operation without [SHIFT] key. And jog operation without shift function (J739) makes it possible to operate only jog operation without [SHIFT] key.

- The spec of the way to move the robot by jog feed differs with spec that is described in “Procedure 33-1 Moving the robot by jog feed” in basic canto of operator manual in case either “Robot operation without shift key function (J591)” or “Jog operation without shift function” is installed.
- The spec of the way of test execution differs with spec that is described in “Procedure 33-2 Step test” and “Procedure 33-3 Continuous test (using the teach pendant)”in basic canto of operator manual in case either “Robot operation without shift key function (J591)” is installed.

Those operations should be replaced with operations that is described following.
Operation without shift function is an optional function.

33.1 JOG OPERATION WITHOUT SHIFT

It is possible to operate jog without [SHIFT] key in case “J591 Robot operation without shift function” or “J739 Jog operation without shift function” is installed to the controller that differs from normal spec. Please submit to following procedure in case you operate jog.

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)

Condition

- The teach pendant must be enabled.
- The system must be in the operation enable state.
- Do not enter the operating area. Do not put any obstacles within the work area.
- Option software “J591 Robot operation without shift” or “J739 Jog operation without shift” is installed.

⚠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 display a desired manual-feed coordinate system on the teach pendant.
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 federate 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 the order to avoid danger, the operator should release the deadman's switch or press the emergency stop button.

5 To move the robot by jog feed, press the jog key correspond to the desired robot motion direction. It is not necessary to press [SHIFT] key. And robot is stopped if all jog keys are released but robot isn't stopped even if [SHIFT] key is released.

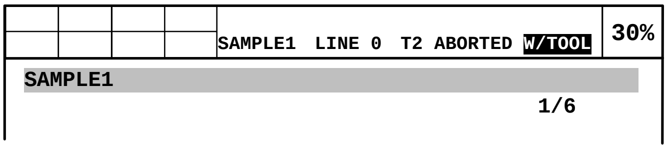
NOTE

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 is displayed.

7 Select 5,TOGGLE WRIST JOG. The mark,[W/],is displayed to show the wrist joint jog mode. To release this mode, select 5,TOGGLE WRIST JOG again.



```
SAMPLE1  LINE 0  T2 ABORTED W/TOOL  30%
SAMPLE1
                                    1/6
```

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.

33.2 TEST EXECUTION OPERATION WITHOUT SHIFT

It is possible to operate test execution without [SHIFT] key in case "J591 Robot operation without shift function" is installed to the controller that differs from normal spec. Please submit to following procedure in case you operate test execution.

Procedure 33-2 Step test (in case robot operation without shift key function is installed)

Condition

- Option software "J591 Robot operation without shift" is installed.
- The teach pendant must be enabled.
- The single-step must be set.
- The system must be in the operation enable state.
- No one must be in the operation area. No obstacle must be placed in the operation area.

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 will be 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, witch may produce unpredictable results. The operator should check that no persons and no unnecessary equipment is in the work area and that each part of the protective fence is sound. Otherwise, injury or property damage would occur. If the program needs to be stopped before it terminates, the operator should release the [FWD]/[BWD] key or deadman switch or press [HOLD] key or emergency stop button.

6 Start the program.
- To perform forward execution of the program, press and hold down [FWD] key, it is not necessary to press [SHIFT] key. Press and hold down [FWD] key until execution of the program is completed. Program is paused if [FWD] key is released during program execution.
- To perform backward execution of the program, press and hold down [BWD] key, it is not necessary to press [SHIFT] key. Press and hold down [BWD] key until execution of the program is completed. Program is paused if [BWD] key is released during program execution.

7 After one line of the program is executed, the program is halted.
- When a motion instruction is executed, the cursor stops at the executed line. The next time forward execution of the program is performed, the next line of the program is executed.
- When a control instruction is executed, the cursor moves to the next line.

8 To release the step mode, press the STEP key.

9 Turn off the teach pendant enable switch, then release the deadman switch.

Procedure 33-3 Continuous test (using the teach pendant) (in case robot operation without shift key function is installed)

Condition

- The teach pendant must be enabled.
- The continuous mode must be set.
- The system must be in the operation enable state.
- No one must be in the operation area. No obstacle must be placed in the operation area.
- Option software “J591 Robot operation without shift” is installed.

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 will be 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, witch may produce unpredictable results. The operator should check that no persons and no unnecessary equipment is in the work area and that each part of the protective fence is sound. Otherwise, injury or property damage would occur. If the program needs to be stopped before it terminates, the operator should release the [FWD] key or deadman switch or press the HOLD or emergency stop button.

6 Press and hold down [FWD] key (it is not necessary to press [SHIFT] key). Hold down [FWD] key until the execution of the program is completed. When [FWD] 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.

34 HIGH SPEED SHIFT KEY FUNCTION

It is possible to operate “High speed SHIFT operation” that is operation that switches speed override to specific value temporary at T1 mode by pressing [SHIFT] key. This function makes it enable to change moving speed of robot high in a moment. It is impossible to use this function at T2 mode.

To use this function, high speed SHIFT key function (J592) is required. In addition, to use high speed shift jog operation, robot operation without shift key function (J591) or jog operation without shift function (J739) is required.

34.1 SETUP FOR HIGH SPEED SHIFT KEY FUNCTION

It is possible to setup the speed override during high speed SHIFT operation. Please input the value of speed override (default value is 0) should be during high speed SHIFT operation to $HSP_SHIFT.$HSP_OVRD. It is possible to input 0-100 but 0 means disable high speed SHIFT function.

NOTE

Speed override is clamped to upper limit in case there is setting upper limit of speed override and value of $HSP_SHIFT.$HSP_OVRD is higher than this value. The speeds at the tool tip and flange surface are limited to 250 mm/sec or slower at T1 mode. So there is no situation that speed exceed 250 mm/sec by this function.

34.2 HIGH SPEED SHIFT JOG OPERATION

It is possible to change moving robot speed high during jog operation by pressing [SHIFT] key. Please refer to following procedure.

NOTE

This operation is operation as its premise of installing option software “J591 Robot operation without shift function” or “J739 Jog operation without shift”, so this operation differs from normal spec.

Procedure 34-1 High speed SHIFT jog operation

Condition

- 3 Mode switch is T1 mode.
- Value of system variable $HSP_SHIFT.$HSP_OVRD is 1-100.
- Speed override isn’t either FINE or VFINE.

Step

1. Hold the teach pendant and press the deadman switch on the back of the teach pendant. Continue pressing the deadman switch during jog feed.
2. Turn on the teach pendant enable switch.
3. Release [SHIFT] key in case [SHIFT] key is pressed and hold.

> ⚠**WARNING**
> Next operation changes the speed of robot moving. Next operation must be executed after you ensure safety. Stop robot moving immediately in case the robot moves with speed that you don’t forecast.

4 Press [SHIFT] key during jog operation. Speed override will be change to value of system variable $HSP_SHIFT.$HSP_OVRD (default value is 100) with back color is changed from green to yellow during [SHIFT] key is being pressed. This operation is called for “high speed SHIFT jog operation”.

> **NOTE**
> It is impossible to operate high speed SHIFT jog operation in case speed override is “FINE” or “VFINE”. Because “FINE” or “VFINE” jog is called inching jog, so these jogs are not sequence jog operation.
> Override changing operation (ex. Pressing override key) is canceled during high speed SHIFT operation.

5 High speed SHIFT operation is end and speed override is restored to the value of override before high speed SHIFT operation if [SHIFT] key is released or jog operation is end during high speed shift operation. But high speed SHIFT operation is end without speed override is restored in case the value of override before high speed SHIFT operation is higher than during high speed SHIFT operation.

34.3 HIGH SPEED SHIFT TEST EXECUTION OPERATION

It is possible to switch speed of robot moving during test execution by pressing [SHIFT] key in only T1 mode. Please refer to following procedure.

> **NOTE**
> This operation is operation as its premise of installing option software “J591 Robot operation without shift function”, so this operation differs with normal spec.

Procedure 34-2 High speed SHIFT test execution operation

Condition

- 3 Mode switch is T1 mode.
- Value of system variable $HSP_SHIFT.$HSP_OVRD is 1-100.

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 will be displayed.
3 Move the cursor to the program start line.
4 Press and hold down the deadman switch, then turn on the teach pendant enable switch.
5 Execute program by Pressing and holding down [FWD]/[BWD] key.
6 Release [SHIFT] key in case [SHIFT] key is pressed and hold.

> ⚠**WARNING**
> Next operation may change the speed of robot moving. Next operation must be executed after you ensure safety. Stop robot moving immediately in case the robot moves with speed that you don’t forecast.

7 Press [SHIFT] key during test execution operation. Speed override will be change to value of system variable $HSP_SHIFT.$HSP_OVRD (default value is 100) with back color is changed from green to

yellow during [SHIFT] key is being pressed. This operation is called for “high speed SHIFT execution operation”.

NOTE

Override changing operation (ex. Pressing override key) is canceled during high speed SHIFT operation.

8 High speed SHIFT operation is end and speed override is restored to the value of override before high speed SHIFT operation if [SHIFT] key or [FWD]/[BWD] key is released during high speed shift operation. But high speed SHIFT operation is end without speed override is restored in case the value of override before high speed SHIFT operation is higher than during high speed SHIFT operation.

35 *i*RDIAGNOSTICS

*i*RDiagnostics function consists of following three functions.

Robot Condition Detection	Warn the users when abnormalities are detected in the reducers.
Servo Diagnosis	Diagnose servo alarms to show possible causes of alarms.
Motion Profiler	Show motion data (OVC, overheat, power consumption) of each program.

35.1 ROBOT CONDTION DETECTION

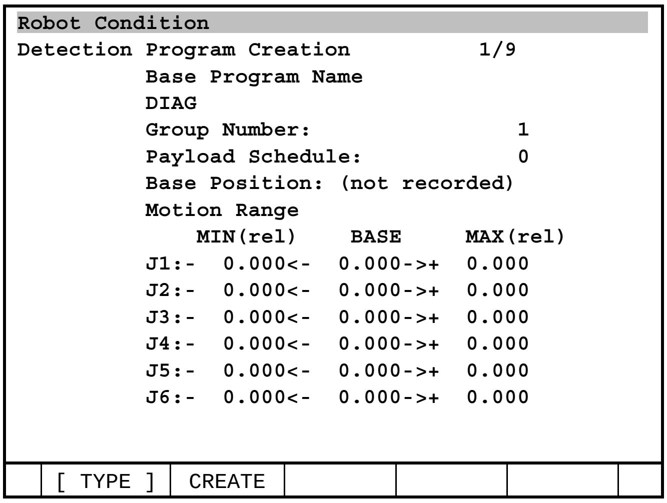
- This function detects abnormalities of reducers to help users decide time for preventive replacement in order to minimize down time due to a problem in reducers.
- By posting warning, it warns the users of abnormality in reducer, detected by periodically analyzing motion data.
- It requires creation of a diagnostic program, registration of initial data, and periodical execution of the diagnostic program.
- The diagnostic program is created by giving base position and allowable motion range.
- Some models do not support this function, contact your local FANUC representative about supported models.
- If enough motion range is not given, motion data cannot be analyzed.
- In case of impulsion-induced rapid degradation, or failure in certain parts, reducer failure may not be detected beforehand as abnormality.
- A failure due to aging may also progress rapidly in a few days before failure. Run the diagnostic program periodically, around once a day.

35.1.1 Setup

“CREATE”

This function requires creation of diagnostic program.
Create diagnostic program with following procedure.

1 Jog the robot to the base position, where diagnostic program starts and ends.
2 Press [MENU] key.
3 Select “UTILITIES”.
4 Press F1, [TYPE].
5 Select “Robot Condition”.



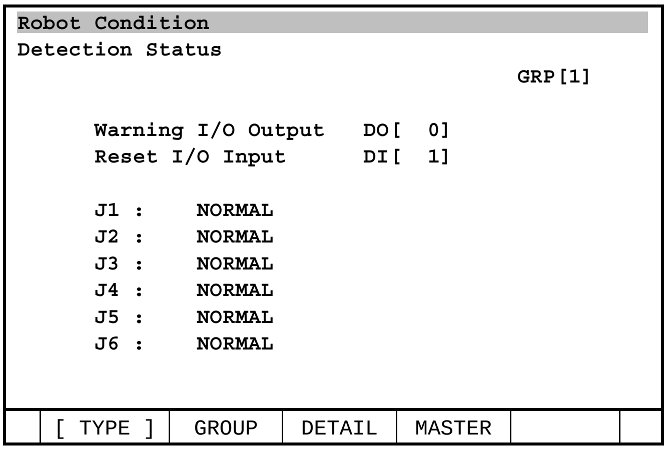
```
Robot Condition
Detection Program Creation          1/9
          Base Program Name
          DIAG
          Group Number:                1
          Payload Schedule:            0
          Base Position: (not recorded)
          Motion Range
              MIN(rel)    BASE     MAX(rel)
          J1:-  0.000<-  0.000->+  0.000
          J2:-  0.000<-  0.000->+  0.000
          J3:-  0.000<-  0.000->+  0.000
          J4:-  0.000<-  0.000->+  0.000
          J5:-  0.000<-  0.000->+  0.000
          J6:-  0.000<-  0.000->+  0.000
```

6 Enter to “Base Program Name” the name of the diagnostic program to be created.
Group number will be added to the base name to create program name.
Please make the base program name not more than 34 letters.
7 Enter to “Group Number” the number of the group for analysis.
8 Enter to "Payload Schedule" the number of the payload setting.
9 Move the cursor over “Base Position”, and press F4, RECORD with [SHIFT] key.
10 Enter allowable motion range of each axis in degrees.
For each axis, enter allowable motion range relative to the base position, both MIN and MAX.
The robot will move axis by axis, one axis at a time.
The axis without wide enough motion range cannot be analyzed.
The axis that is set motion range of 0 degree will not be analyzed.
11 Set to AUTO mode and disable TP.
12 Set override to 100%.
13 Press F2, CREATE with [SHIFT] key.

A program will be created with group number added to the Base Program Name.

NOTE
Please run the program with low override to assure safety of the program.

“MASTER”
Next, follow the procedure for the registration of the initial data.
1 Run the created program with 100% override (run with low override beforehand to assure safety).
2 Press [MENU] key.
3 Select “STATUS”.
4 Press F1, [TYPE].
5 Select “Robot Condition”.



```
Robot Condition
Detection Status
                                          GRP[1]

      Warning I/O Output   DO[  0]
      Reset I/O Input      DI[  1]

      J1 :    NORMAL
      J2 :    NORMAL
      J3 :    NORMAL
      J4 :    NORMAL
      J5 :    NORMAL
      J6 :    NORMAL

   [ TYPE ]   GROUP   DETAIL   MASTER
```

6 If upper right display of GRP[] does not show the group for analysis, press F2, GROUP and enter the group number.
7 Press F4, MASTER key.
8 Enter –1 (all axes) when prompted to enter the axis number.
9 A message will ask to confirm removal of previous data and use of current data as base data.
“Please confirm to use current data as baseline data? Please note previous history will be removed after master.”
Select [YES].

The latest analysis data is registered as base line data.
The axis not registered, shown with *s was not given enough motion range to analyze the condition.

NOTE

Please note the followings, for they may prevent the function from working properly.

- Do not edit the created program.
- Creating the program with the same name will overwrite the old program. If overwritten after “MASTER,” the function does not work properly. If the program needs to be changed during setup, go through both “CREATE” and “MASTER”.
- Different programs may be created with arbitrary base positions. However, an axis can be analyzed with only one program. With different programs analyzing the same axis, the function does not work properly because of different characteristics of the data collected. When tried to analyze an axis with a program different from the one used for “MASTER,” following warning will be posted.
  "DIAG-008 Baseline prog changed (G:g A:a)"
  where g is the group number and a is the axis number.
- Basically, go through “MASTER” only with initial setup and when replacing reducer.
- If it becomes impossible to use the program for some reason, such as moving the system, it would be necessary to go through “CREATE” and “MASTER”. However, if there is a reducer already reaching an end of its operating life, the function may not work properly.

35.1.2 Execute Program

After the base line data is registered, the latest data is analyzed every time diagnostic program is executed with 100% override, and check if there are any abnormalities.
Run the program periodically to detect abnormality: for example, once a day.
Make sure that the program is run under the same condition as the time the baseline data was recorded, such as load.
In the case that the software version is 7DC1/01-18 or 7DC2/01-04, make sure that the UTOOL number is set to1 before executing the program.

If abnormality is detected, following warning will be posted,
“SRVO-394 Reducer Abnormal (Gg, Aa)”
where g is the group number and a is the axis number.
Abnormality is detected in corresponding axis. Please plan replacement.

35.1.3 Robot Condition Detection Status

Current condition can be viewed at Robot Condition Detection Status.
To view Robot Condition Detection Status,

1 Press [MENU] key.
2 Select “STATUS”.
3 Press F1, [TYPE] key.
4 Select “Robot Condition”.

Robot Condition Detection Status shows condition of each axis.
Moving the cursor over an axis and pressing F3, DETAIL will show the time of registration of the base line data and time of analysis of the latest data.

If "Warning I/O Output" DO is set, the specified DO turns ON when abnormality is detected.
If "Reset I/O Input" DI is set, the "Warning I/O Output" DO is cleared (i.e. turns OFF) when the specified DI turns ON.

35.1.4 After Replacement

When reducer is replaced, corresponding base line data need to be re-registered.
Follow the procedure to re-register the data.

1. Run the diagnostic program with 100% override. (Run with low override beforehand to assure safety.)
2. Press [MENU] key.
3. Select “STATUS”.
4. Press F1, [TYPE].
5. Select “Robot Condition”.
6. If upper right display of GRP[] does not show the group for analysis, press F2, GROUP and enter the group number.
7. Move the cursor over the axis of replacement, and press F3, DETAIL.
8. Check that the time of the Latest is the time diagnostic program was run in step 1.
9. Press F4 MASTER.
10. Enter the axis of replacement when prompted to enter the axis number.
11. A message will ask to confirm removal of previous data and use of current data as base data.
    “Please confirm to use current data as baseline data? Please note previous history will be removed after master.”
    Select [YES].
12. Press F3, DETAIL and check that time of the Base is updated.

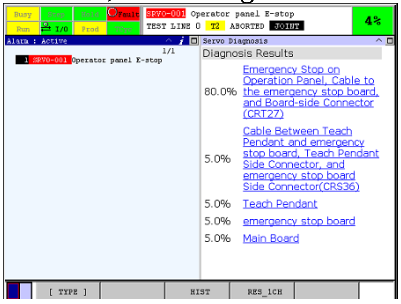
The new base line data is registered.

35.2 SERVO DIAGNOSIS

- This function diagnoses alarms posted, and list probable causes, aiming to decrease down time before resetting alarm.
  - ➢ This function estimates where the problem is, based on the alarm posted and internal information acquirable by software. However, it may not be able to estimate the problem in some cases such as if problem rises where software cannot acquire information. Please use it to help solve problem when alarm is posted.
- When servo alarm is posted, follow the procedure to perform Servo Diagnosis.

(Alarm Posted.)

1. Press [MENU] key.
2. Select “ALARM”. “Alarm: Active” would be displayed.
3. Press [FCTN] key with [i] key
4. Select "Servo Diagnosis" in the "Related Views" menu. The causes of the alarm would be listed. Please check the items to see if there is any trouble.
   With software versions 7DC2/16 or later, Servo Diagnosis can also be performed by F5 “DIAG” key.



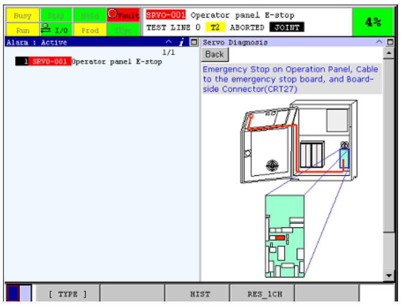
SRVO-001 Operator panel E-stop
TEST LINE 0 T2 ABORTED JOINT

1/1
1 SRVO-001 Operator panel E-stop

Diagnosis Results

80.0%	Emergency Stop on Operation Panel, Cable to the emergency stop board, and Board-side Connector (CRT27)
5.0%	Cable Between Teach Pendant and emergency stop board, Teach Pendant Side Connector, and emergency stop board Side Connector(CRS36)
5.0%	Teach Pendant
5.0%	emergency stop board
5.0%	Main Board

5. Some of the causes text have a hyperlink. As the hyperlink is tapped, the diagram about the cause is shown. (7DC2 or later)



SRVO-001 Operator panel E-stop
TEST LINE 0 T2 ABORTED JOINT

1/1
1 SRVO-001 Operator panel E-stop

Emergency Stop on Operation Panel, Cable to the emergency stop board, and Board-side Connector(CRT27)

Click “Back” button to return back to the cause list view.

- The alarms covered by this function are following:
  - SRVO-001 Operator panel E-Stop
  - SRVO-005 Robot overtravel
  - SRVO-006 Hand broken
  - SRVO-009 Pneumatic pressure alarm
  - SRVO-018 Brake abnormal (G:%d A:%d)
  - SRVO-021 SRDY off (Group:%d Axis:%d)
  - SRVO-022 SRDY on (Group:%d Axis:%d)
  - SRVO-023 Stop error excess (G:%d A:%d)
  - SRVO-024 Move error excess (G:%d A:%d)
  - SRVO-036 Inpos time over (G:%d A:%d)
  - SRVO-038 Pulse mismatch (G:%d A:%d)
  - SRVO-043 DCAL alarm (Group:%d Axis:%d)
  - SRVO-044 DCHVAL alarm (Group:%d Axis:%d)
  - SRVO-045 HCAL alarm(Group:%d Axis:%d)
  - SRVO-046 OVC alarm (Group:%d Axis:%d)
  - SRVO-047 LVAL alarm(Group:%d Axis:%d)
  - SRVO-049 OHAL1 alarm (Grp:%d Ax:%d)
  - SRVO-050 Collision Detect alarm (G:%d A:%d)
  - SRVO-051 CUER alarm(Group:%d Axis:%d)
  - SRVO-055 FSSB com error 1 (G:%d A:%d)
  - SRVO-056 FSSB com error 2 (G:%d A:%d)
  - SRVO-057 FSSB disconnect (G:%d A:%d)
  - SRVO-062 BZAL alarm(Group:%d Axis:%d)
  - SRVO-064 PHAL alarm(Group:%d Axis:%d)
  - SRVO-067 OHAL2 alarm (Grp:%d Ax:%d)
  - SRVO-068 DTERR alarm (Grp:%d Ax:%d)
  - SRVO-069 CRCERR alarm (Grp:%d Ax:%d)
  - SRVO-070 STBERR alarm (Grp:%d Ax:%d)
  - SRVO-071 SPHAL alarm (Grp:%d Ax:%d)
  - SRVO-072 PMAL alarm(Group:%d Axis:%d)
  - SRVO-073 CMAL alarm(Group:%d Axis:%d)
  - SRVO-074 LDAL alarm(Group:%d Axis:%d)
  - SRVO-076 Tip Stick Detection(G:%d A:%d)
  - SRVO-105 Door open or E.Stop
  - SRVO-136 DCLVAL alarm (G:%d A:%d)
  - SRVO-156 IPMAL alarm (G:%d A:%d)
  - SRVO-157 CHGAL alarm (G:%d A:%d)
  - SRVO-204 External(SVEMG abnormal) E-stop
  - SRVO-205 Fence open(SVEMG abnormal)
  - SRVO-206 Deadman switch (SVEMG abnormal)
  - SRVO-213 E-STOP Board FUSE2 blown
  - SRVO-214 6ch amplifier fuse blown(R:%d)
  - SRVO-215 Brake Unit fuse blown(R:%d)

- SRVO-216 OVC(total) (%d)
- SRVO-233 TP OFF in T1,T2/Door open
- SRVO-235 Short term Chain abnormal
- SRVO-251 DB relay abnormal(G:%d A:%d)
- SRVO-252 Current detect abnl(G:%d A:%d)
- SRVO-253 Amp internal over heat(G:%d A:%d)
- SRVO-277 Panel E-stop(SVEMG abnormal)
- SRVO-278 TP E-stop(SVEMG abnormal)
- SRVO-290 Dclink HC alarm(G:%d A:%d)
- SRVO-291 IPM over heat (G:%d A:%d)
- SRVO-292 EXT.FAN alarm (G:%d A:%d)
- SRVO-349 DCS MCC ON alarm %x,%x

35.3 MOTION PROFILER

Motion Profiler is a motion option providing important motion information for either total execution cycle or each executed program line. This option enables the easy diagnosis of program performance for selected program. The exact motion lines can be identified and then adjusted easily using this option so the desired performance can be achieved.
The summary data for execution cycle provided by this option includes:

- Total Cycle Time
- Reducer Load Monitor (max. value)
- Joint Load Monitor (intended models only, max. value)
- M430 Load Monitor (intended models only, if warning is posted)
- Reducer Estimated Life (L10 life)
- Power Consumption
- Regenerative Power (if hardware is available)
- Steady-State OVC (the prediction if the same path running non-stop continuously)
- Overheat

The detailed data shown in bar charts per program execution line includes:

- Execution Time per line
- Max. value of Reducer Load Monitor per line
- Power usage per line

Due to the memory constraints of controllers, the motion profiler data for the extreme long program may be overwritten and data may be missing. Please consider separating the very long program to reasonable program sizes.

35.3.1 Setup

Set up Motion Profiler with following procedure.

1 Press [MENU] key.
2 Select “STATUS”.
3 Press F1, [TYPE].
4 Select “Motion Profiler”.

With software prior to 7DC1/06, please set up Motion Profiler with following procedure.

1 Press [MENU] key.
2 Select “SETUP”.
3 Press F1, [TYPE].
4 Select “Motion Profiler”.

Motion profiler function will be turned off automatically after the selected program is aborted. Please enable the function again if the program needs to be analyzed again.

It is important that ambient temperature, special hardware option, and robot connection cable (RCC) length for each robot group are entered correctly. The prediction and calculation result will be affected by these settings.

Table 35.3.1 Menu items and descriptions for motion profiler's setup menu

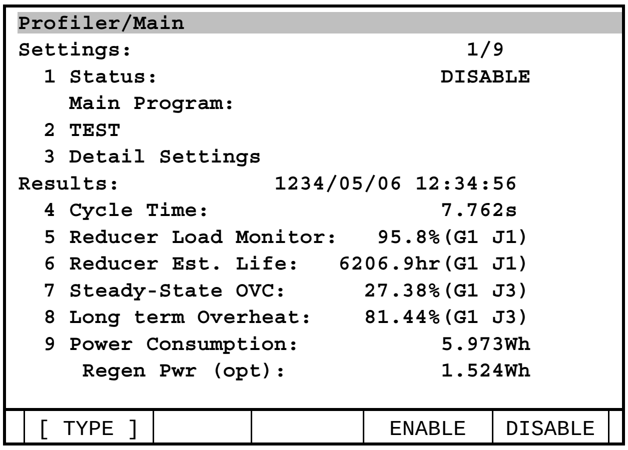
Item		Description
Status		Enable or disable Motion Profiler. This status will change to disable automatically after the main program is aborted.
Main Program		The main TP program to be used for Motion Profiler
Detail Settings	Group Number	Group number
	Ambient Temperature	The ambient temperature for this robot
	Robot Connection Cable Length	The length of robot connection cable (RCC) for this robot This number may be printed on the robot connection cable.
	Hardware Configuration	Any special hardware configuration for this robot. For example, motor fan, motor cover, etc.

35.3.2 Results

The results of Motion Profiler function can be reviewed in either Status menu or directly go to the "related view" from TP editor.

- Status screen

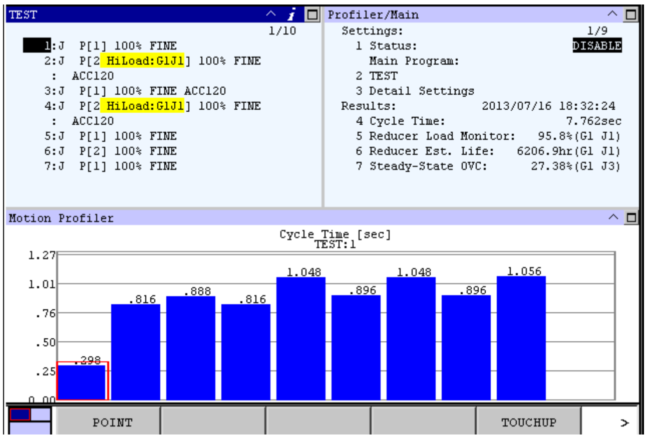
1 Press [MENU] key.
2 Select "STATUS".
3 Press F1, [TYPE].
4 Select "Motion Profiler" to show Main page.
5 Select data and press F3, DETAIL to display the detail bar chart with program name: line number as horizontal axis, or data of each axis.



```
Profiler/Main
Settings:                          1/9
  1 Status:                      DISABLE
    Main Program:
  2 TEST
  3 Detail Settings
Results:            1234/05/06 12:34:56
  4 Cycle Time:                   7.762s
  5 Reducer Load Monitor:    95.8%(G1 J1)
  6 Reducer Est. Life:   6206.9hr(G1 J1)
  7 Steady-State OVC:      27.38%(G1 J3)
  8 Long term Overheat:    81.44%(G1 J3)
  9 Power Consumption:          5.973Wh
     Regen Pwr (opt):           1.524Wh
```

- Related view

1 Press [SELECT] key.
2 Select TP program that Motion Profiler was enabled and press Enter.
3 In TP program editor, press [i] + [FCTN] key to display Related View menu.
4 Select "Motion Profiler" to review triple pane with TP editor, Motion Profiler Main, and Motion Profiler detailed chart. Scrolling the cursor in the TP editor then Motion Profiler detailed chart will update to current line data automatically. This will work in the sub program as well.



```
TEST                                    1/10
   1:J  P[1] 100% FINE
   2:J  P[2 HiLoad:G1J1] 100% FINE
    :  ACC120
   3:J  P[1] 100% FINE ACC120
   4:J  P[2 HiLoad:G1J1] 100% FINE
    :  ACC120
   5:J  P[1] 100% FINE
   6:J  P[2] 100% FINE
   7:J  P[1] 100% FINE
```

```
Profiler/Main
 Settings:                             1/9
   1 Status:                       DISABLE
     Main Program:
   2 TEST
   3 Detail Settings
 Results:              2013/07/16 18:32:24
   4 Cycle Time:                  7.762sec
   5 Reducer Load Monitor:   95.8%(G1 J1)
   6 Reducer Est. Life:   6206.9hr(G1 J1)
   7 Steady-State OVC:       27.38%(G1 J3)
```

Unsupported axes or data:

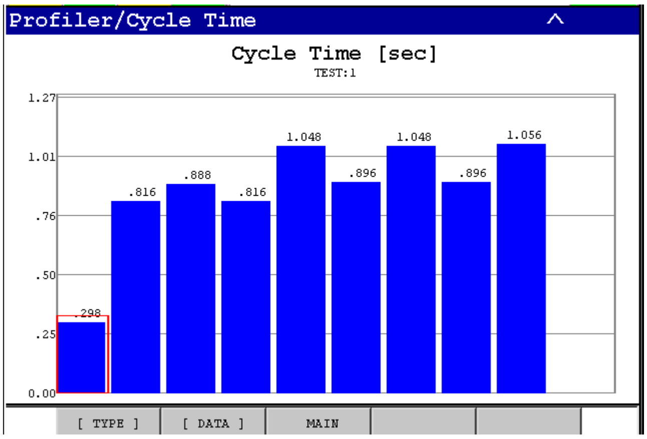
If there are unsupported axes or data, next to the “Results:” will be displayed (exclusions). Selecting (exclusions) will show the list of unsupported axes and/or data. (7DC1/09 or later)

35.3.2.1 Detail

Main page displays summarized data of analyzed program. With cursor over each item, pressing F3, DETAIL shows detailed data of each item.

[Cycle Time]

Shows execution time of the analyzed program. In DETAIL, bar chart shows execution time of each program line. The lines with very short execution time will not be shown, including some of command lines.

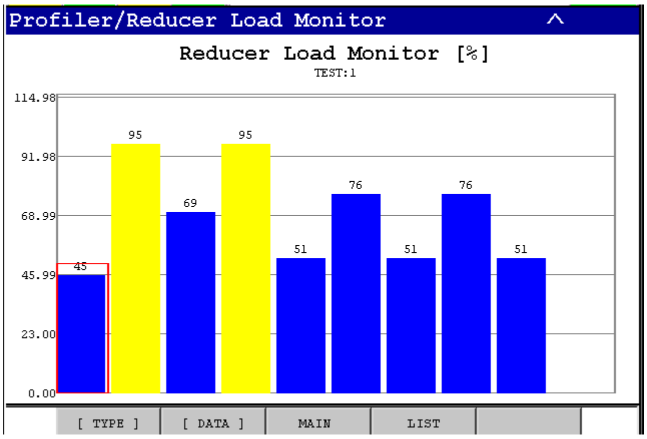


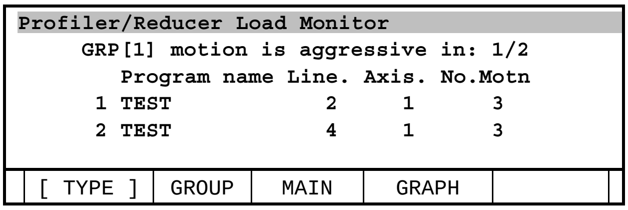
[Reducer Load Monitor] (shown only for supported models)

Calculates load level to drive element, such as reducer or gear, from motion command. Continued use over 100% of load level may result in premature failure of driving element, such as reducer or gear. If the load level exceeds 100%, change the program, such as acceleration override instruction (ACC) or feed rate, so that the load level is below 100%.

Main page displays maximum value within the analyzed program, and the group and axis. In DETAIL, bar chart shows maximum values per line. Lines with aggressive motion will be colored yellow and red. The yellow bar shows that the load level is near the limit, and the red bar shows that the load level exceeds the limit. In LIST, it shows list of aggressive motions with program name, line number, axis number, and number of motion. If motion command of aggressive motion is derived from more than one line, number of motion shows number of lines involved. After the analysis, if there are aggressive motions, corresponding line will have comment “HiLoad:G(group number)” added when EDIT is displayed with TP enabled.

When load level is over 100%, “DIAG-005 Reducer load excess” warning would be posted in addition to the display in this option.





```
Profiler/Reducer Load Monitor
       GRP[1] motion is aggressive in: 1/2
          Program name Line. Axis. No.Motn
        1 TEST             2     1      3
        2 TEST             4     1      3
```

[Joint Load Monitor] (For certain models only. Shown only for supported models)

Displays load level to joints. If it exceeds 100%, “MOTN-522 Joint load excess” alarm will stop the robot. There is no DETAIL.

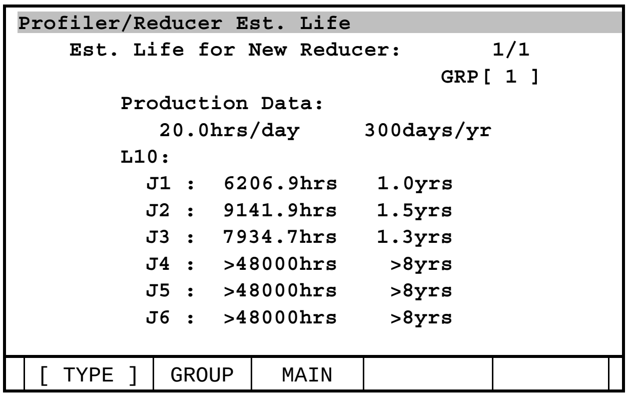
[M430 Load Monitor] (For certain models only. Shown only for supported models)

Calculates load to reducers in different way from Reducer Load Monitor. If “warning” is displayed, it means that “MOTN-520 J(number of axis)TqOver” is posted. Change the program in the same way as Reducer Load Monitor so that the warning would not be posted. There is no DETAIL.

[Reducer Estimated Life] (Some axes are not targeted. Shown only for supported models)

Displays L10 life of reducers if analyzed program is executed repeatedly using new reducers. Main page shows the shortest L10 life, its group and axis. In DETAIL, L10 life of each axis is shown. Also in DETAIL, production data, operating hours per day and operating days per year, which would be used for L10 life calculation, ca be changed.

L10 life is the time after which 10% of reducers fail. Evaluate the L10 life in the context of intended use.



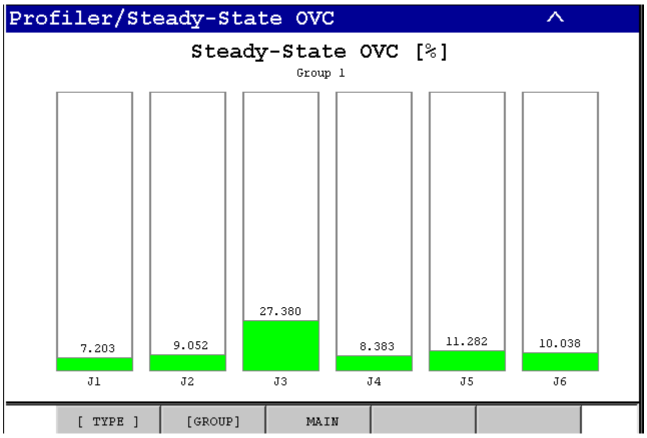
```
Profiler/Reducer Est. Life
    Est. Life for New Reducer:       1/1
                                  GRP[ 1 ]
        Production Data:
           20.0hrs/day     300days/yr
        L10:
          J1 :  6206.9hrs   1.0yrs
          J2 :  9141.9hrs   1.5yrs
          J3 :  7934.7hrs   1.3yrs
          J4 :  >48000hrs    >8yrs
          J5 :  >48000hrs    >8yrs
          J6 :  >48000hrs    >8yrs
```

NOTE

Reducer Load Monitor, Joint Load Monitor, M430 Load Monitor, and Reducer Estimated Life require that the payload is set correctly. Otherwise, they do not work properly.
Also, the calculations are done with the assumption that there is no contact between robot and other objects. Therefore, a program that includes contact and causes external force (such as pushing a work piece while mounting) will not be analyzed properly.

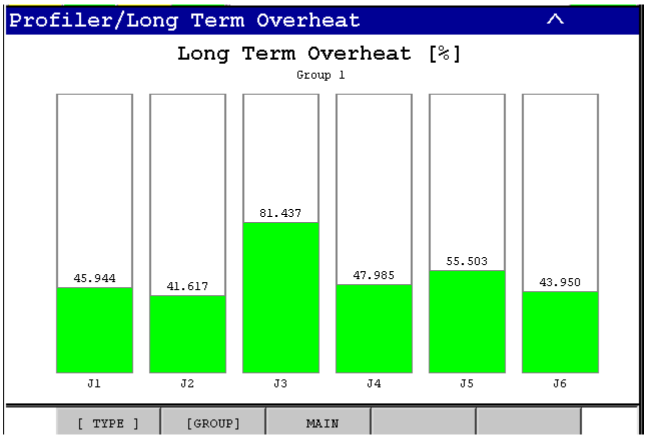
[Steady-State OVC]

Shows estimation of OVC value(The ratio of the motor temperature simulated by the software to the alarm threshold) when analyzed program was run repeatedly. In Main, the largest value among the axes is shown. In DETAIL, the values of each axis are shown. The value over 100% shows estimated result of "SRVO-046 OVC alarm (Group:i Axis:j)" occurring.



[Overheat]

Actual overheat alarm(SRVO-067 OHAL2 alarm (Grp:i Ax:j)) is triggered by sensor in the motor, but this function estimates if overheat alarm would occur, by comparing electric current data to experimental data. In Main, the largest value among the axes is shown. In DETAIL, the values of each axis are shown. The values represent the ratio of calculated value against the thresholds. The value over 100% shows estimated result of overheat alarm occurring. The estimation is affected by Ambient Temperature and Hardware Configuration set in Detail Settings. Please set them accurately.

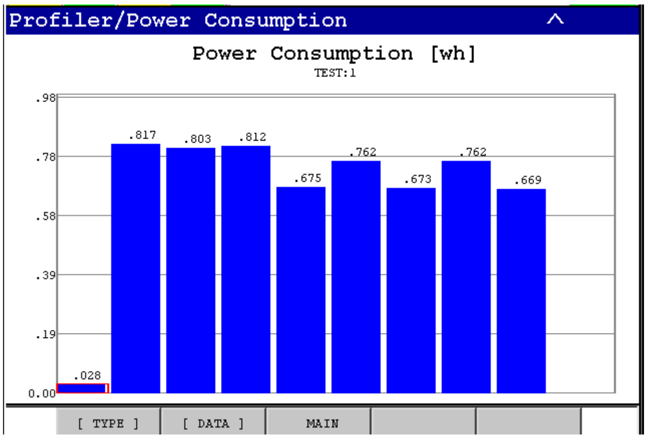


NOTE

Steady-State OVC and Overheat are results of estimation, and their accuracies are not guaranteed. Please use them only as guides.

[Power Consumption]

Shows power consumption while executing the analyzed program. In DETAIL, power consumption per axis is shown. They are estimated results and not measured data. In cases such as unsupported group included in motion groups of the program, the results do not include their power consumption. The results are affected by Robot Connection Cable Length set in Detail Settings. Please set it accurately.



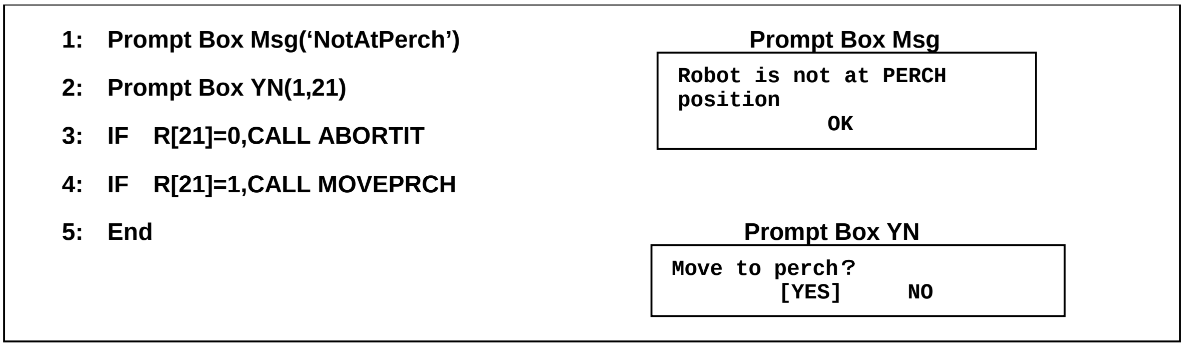
[Regenerative Power]

Shows the estimation of regenerated energy while executing the analyzed program, if the hardware option is installed. The result is affected by Robot Connection Cable Length set in Detail Settings. Please set it accurately. There is no DETAIL.

36 MENU UTILITY FUNCTION

You can create your own teach pendant menus to help you application run more smoothly. To use your own menus, first perform menu setup then call the menu macro program from a teach pendant program and select the desired menu in the parameter.

See Fig. 36 for an example program and example menus.



```
1:  Prompt Box Msg('NotAtPerch')
2:  Prompt Box YN(1,21)
3:  IF  R[21]=0,CALL ABORTIT
4:  IF  R[21]=1,CALL MOVEPRCH
5:  End
```

Prompt Box Msg

```
Robot is not at PERCH
position
            OK
```

Prompt Box YN

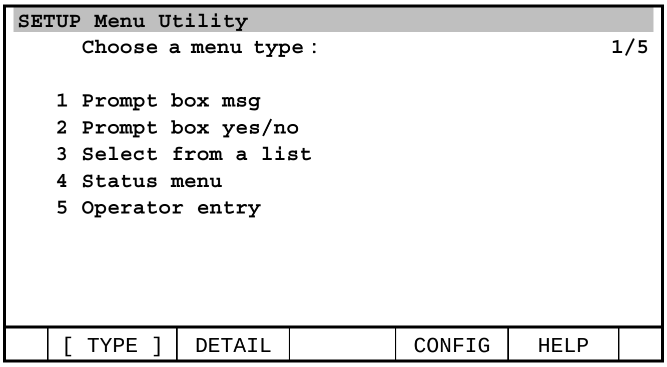
```
Move to perch?
        [YES]     NO
```

Fig.36 Example for teach pendant program that uses menu utility function

Above example is in case position of robot isn't in safe area. If macro program in line 1 "Prompt box Msg('NotAtPerch')" is implemented, prompt box message screen is displayed so that to make sure the operator is aware of something. And if macro program in line 2 "Prompt Box YN(1.21)" is implemented, prompt box yes/no screen is displayed, so that the teach pendant program pauses until the operator selects YES or NO and presses ENTER. In this case, teach pendant program "MOVEPRCH" is called by line 3 is implemented if YES is selected and teach pendant program "ABORT" is called by line 4 is implemented if NO is selected. In this way prompt box message or menu is displayed so that it is possible to prompt the operator to be caution or let the operator do work smoothly.

36.1 SETUP ABOUT MENU UTILITY

If you press MENU and select "6 SETUP" and select "Menu Utility", the following menu utility display will be displayed. There are five menu type "Prompt box msg", "Prompt box yes/no", "Select from a list", "Status menus" and "Operator entry".



```
SETUP Menu Utility
     Choose a menu type:                          1/5

   1 Prompt box msg
   2 Prompt box yes/no
   3 Select from a list
   4 Status menu
   5 Operator entry
```

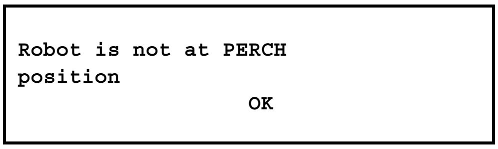
Each menu type corresponds with macro program as following table 1, so each behavior of macro program will be changed if each setting of menu type is changed.

Table 36.1 Description for each menu utility screen item

ITEM	Macro Program	DESCRIPTION
Prompt box msg	Prompt Box Msg (Menu)	This item is used to make sure the operator is aware of something. When a prompt box message screen is displayed, the teach pendant program pauses until the operator pressed ENTER.
Prompt box yes/no	Prompt Box YN (menu,answer_reg)	This item asks the operator a question that can be answered YES or NO. When a prompt box yes/no screen is displayed, the teach pendant program pauses until the operator selects YES or NO and presses ENTER.
Select from a list	List Menu(menu,answer_reg)	This item asks the operator to select an item from a list of items. When a list menu is displayed, the teach pendant program pauses until the operator selects an item form the list.
Status menus	Status Menu(menu)	This item displays status information to the teach pendant screen. When a status menu is executed, the teach pendant displays the status menu and then continues executing. The values that are displayed are refreshed at least once every second.
Operator entry	Op. Entry Menu(menu)	This item requires the operator to enter INTEGER,REAL,BOOLEAN, or text values. When an operator entry menu is executed, the program pauses until the operator types the information and then presses F2 DONE.

36.1.1 Prompt Box Msg

A prompt box message menu is used to make sure the operator is aware of something. When a prompt box message screen is displayed, the teach pendant program pauses until the operator presses ENTER. In addition, when the prompt box is displayed, the operator cannot access any other menus.



```
Robot is not at PERCH
position
                  OK
```

Macro: Prompt Box Msg (menu)
A prompt box message menu uses the Prompt Box Msg(menu) macro.

Parameter 1: menu
This is the menu number or menu name. The menu can be chosen by number or name.

Usage example:

Example 1 (setting menu number to parameter):

```
PROG   TEST1
    1: Prompt Box Msg(2)
```

Example 2 (setting menu name to parameter):

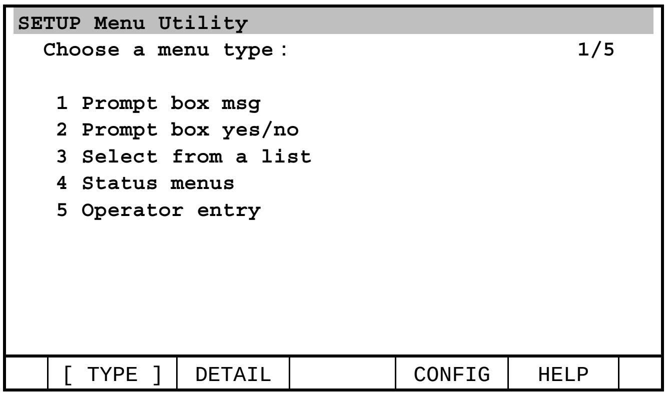
```
PROG   TEST2
    1: Prompt Box Msg('NotAtPerch')
```

Save Menu setting:
Each prompt box message menu is stored in its own variable file that begins with the prefix “MENU1”. The next two characters of the filename are the menu number. For example, the prompt box message menu 1 is stored in the variable file named “MENU101.VR”.

Procedure 36-1 Setup Prompt Box Msg

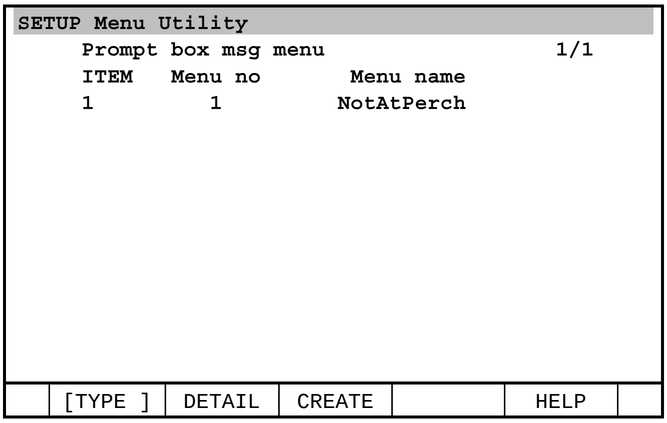
Step

1 Press [MENU] key and select ”6 SETUP” and select ”Menu Utility” so that Menu utility screen will be displayed.



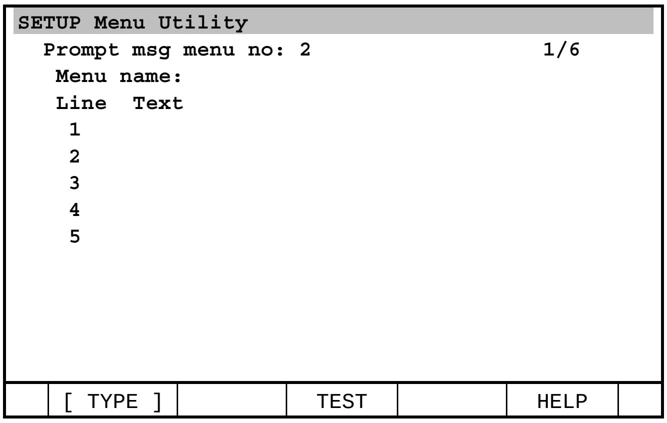
```
SETUP Menu Utility
  Choose a menu type :                        1/5

   1 Prompt box msg
   2 Prompt box yes/no
   3 Select from a list
   4 Status menus
   5 Operator entry

  [ TYPE ]  DETAIL          CONFIG    HELP
```

2 Set cursor to “1 Prompt box msg” and press F2, DETAIL so that Prompt box msg menu will be displayed. You will see a screen similar to the following.



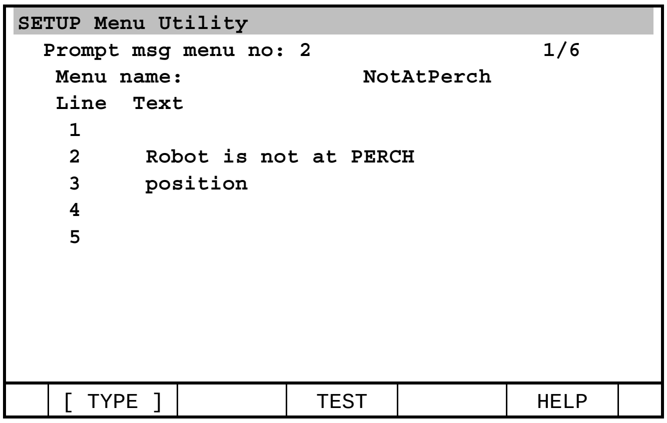
```
SETUP Menu Utility
     Prompt box msg menu                     1/1
     ITEM   Menu no       Menu name
     1         1         NotAtPerch

  [TYPE ]  DETAIL  CREATE            HELP
```

3 Press F3, CREATE if you want to create new menu. Press F2, DETAIL if you want to change menu that already exists. You will see a screen similar to the following.



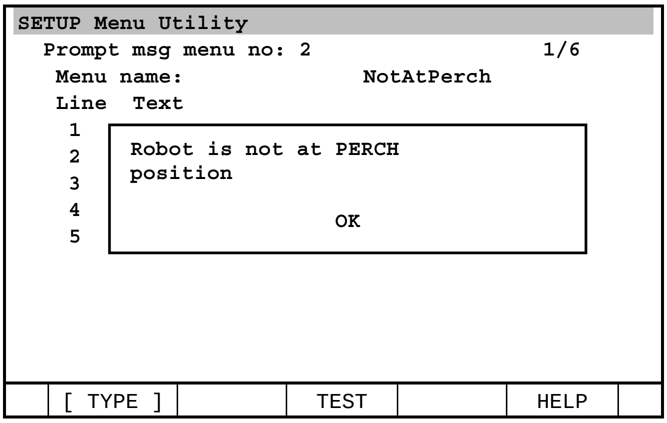
```
SETUP Menu Utility
  Prompt msg menu no: 2                      1/6
   Menu name:
   Line  Text
    1
    2
    3
    4
    5

  [ TYPE ]          TEST              HELP
```

4 Input menu name and message that you want to display. (ex. Setting menu no.2 as menu name “NotAtPerch”)



```
SETUP Menu Utility
  Prompt msg menu no: 2                  1/6
   Menu name:              NotAtPerch
   Line  Text
    1
    2     Robot is not at PERCH
    3     position
    4
    5
```

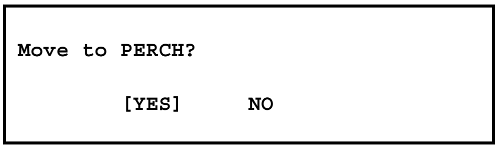
5 Press F3, TEST if you want to confirm the message that will be displayed. You will see a screen similar to the following.



```
SETUP Menu Utility
  Prompt msg menu no: 2                  1/6
   Menu name:              NotAtPerch
   Line  Text
    1
    2   Robot is not at PERCH
    3   position
    4
    5                 OK
```

36.1.2 Prompt Box Yes/No Menu

A prompt box yes/no menu asks the operator a question that can be answered YES or NO. When a prompt box yes/no screen is displayed, the teach pendant program pauses until the operator selects YES or NO and presses ENTER. In addition, when the prompt box yes/no screen is displayed, the operator cannot access any other menus.



```
Move to PERCH?

        [YES]     NO
```

Macro: Prompt Box YN (menu, answer_reg)
A prompt box yes/no menu uses the Prompt Box YN(menu, answer_reg) macro.

Parameter 1: menu
This is the menu number or menu name. The menu can be chosen by number or name.

Parameter 2: answer_reg
This is the register that will contain the answer from the operator.

Usage example:
Example 1 (setting menu number to parameter):

```
PROG   TEST1
   1: Prompt Box YN(2, 2)
```

```
2: IF R[2] = 0, CALL ABORTIT
```

Example 2 (setting menu name to parameter):

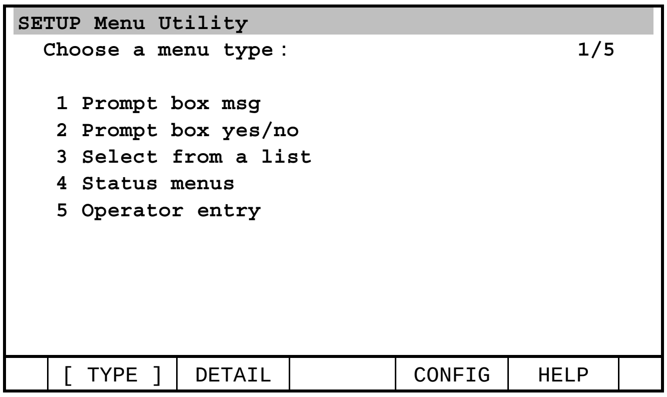
```
PROG   TEST2
   1: Prompt Box YN('IsItSafe', 6)
   2: IF R[6] = 0, JMP LBL[5]
```

Save Menu setting:
Each prompt box yes/no menu is stored in its own variable file that begins with the prefix “MENU2”. The next two characters of the filename are the menu number. For example, the prompt box message menu 1 is stored in the variable file named “MENU201.VR”.

Procedure 36-2 Setup Prompt Box YN

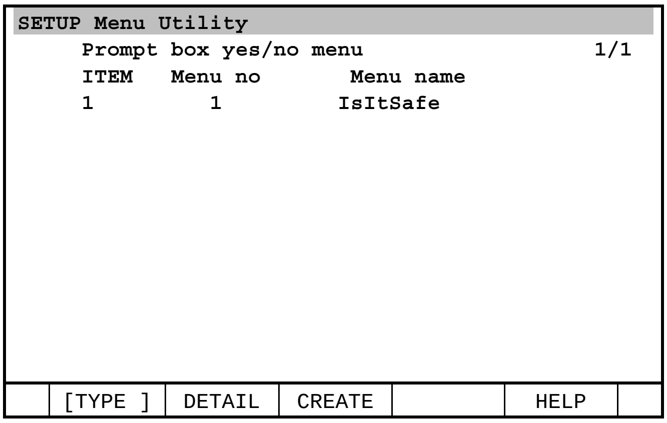
Step

1 Press [MENU] key and select ”6 SETUP” and select “Menu Utility” so that Menu utility screen will be displayed.



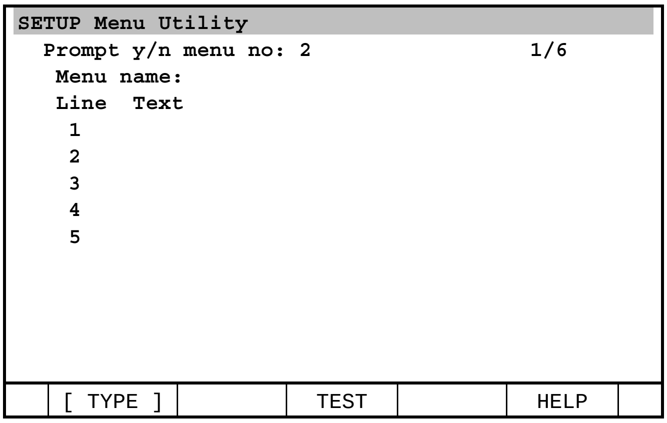
```
SETUP Menu Utility
  Choose a menu type :                      1/5

   1 Prompt box msg
   2 Prompt box yes/no
   3 Select from a list
   4 Status menus
   5 Operator entry
```

2 Set cursor to “2 Prompt box yes/no” and press F2, DETAIL so that Prompt box yen/no menu will be displayed. You will see a screen similar to the following.



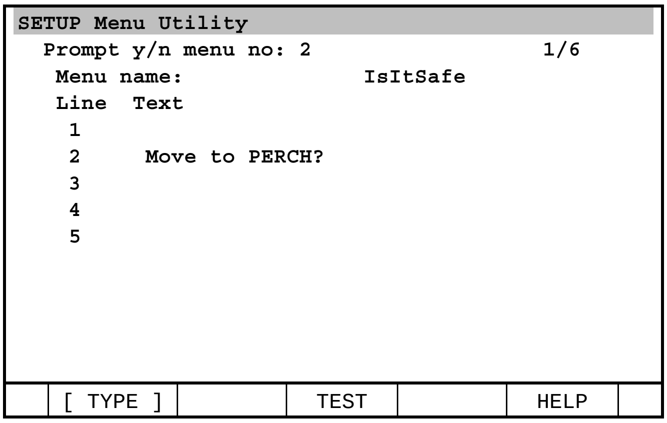
```
SETUP Menu Utility
     Prompt box yes/no menu                  1/1
     ITEM   Menu no       Menu name
     1         1         IsItSafe
```

3 Press F3, CREATE if you want to create new menu. Press F2, DETAIL if you want to change menu that already exists. You will see a screen similar to the following.



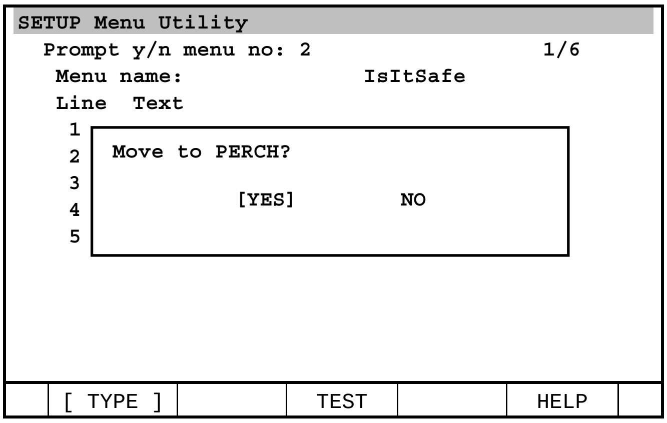
```
SETUP Menu Utility
  Prompt y/n menu no: 2                  1/6
   Menu name:
   Line  Text
    1
    2
    3
    4
    5

   [ TYPE ]         TEST           HELP
```

4 Input menu name and message that you want to display. (ex. Setting menu no.2 as menu name “IsItSafe”)



```
SETUP Menu Utility
  Prompt y/n menu no: 2                   1/6
   Menu name:              IsItSafe
   Line  Text
    1
    2     Move to PERCH?
    3
    4
    5

   [ TYPE ]         TEST           HELP
```

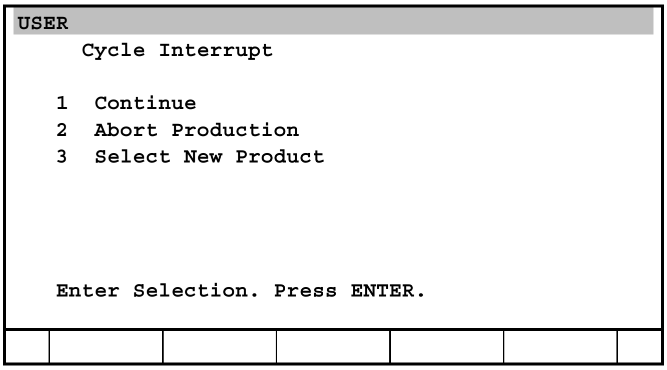
5 Press F3, TEST if you want to confirm the message that will be displayed. You will see a screen similar to the following.



```
SETUP Menu Utility
  Prompt y/n menu no: 2                   1/6
   Menu name:              IsItSafe
   Line  Text
    1
    2  Move to PERCH?
    3
    4            [YES]        NO
    5

   [ TYPE ]         TEST           HELP
```

36.1.3 List Menu

A list menu asks the operator to select an item from a list of items. When a list menu is displayed, the teach pendant program pauses until the operator selects an item from the list.



```
USER
     Cycle Interrupt

   1  Continue
   2  Abort Production
   3  Select New Product




   Enter Selection. Press ENTER.
```

Macro: List Menu(menu,answer_reg)
A list menu uses the List Menu(menu,answer_reg) macro.

Parameter 1: menu
This is the menu number or menu name. The menu can be chosen by number or name.

Parameter 2: answer_reg
This is the register that will contain the answer from the operator.

Usage example:
Example 1:

```
 PROG   TEST1
     1: List Menu (2,21)
     2: SELECT R[21] = 1, CALL TOOL1
     3:              = 2, CALL TOOL2
     4:              = 3, JMP LBL[2]
     5:              ELSE, CALL ABORTIT
```

Example 2:

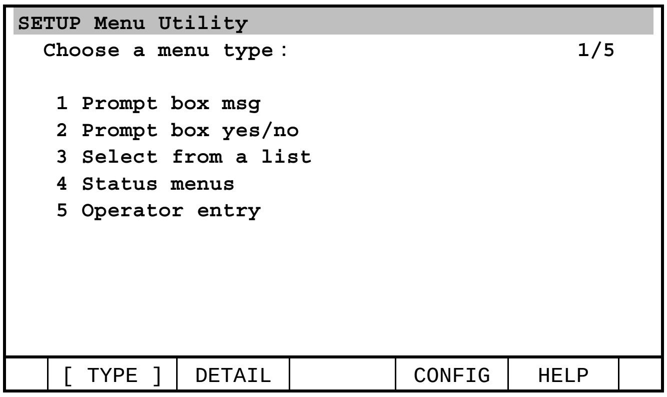
```
 PROG   TEST2
     1: List Menu (‘ErrorRecov’,2)
     2: SELECT R[2] = 1, JMP LBL[2]
     3:             = 2, CALL MOVEMANT
     4:             = 3, CALL MOVEPRCH
     5:             ELSE, CALL ABORTIT
```

Save Menu setting:
Each list menu is stored in its own variable file that begins with the prefix “MENU3”. The next two characters of the filename are the menu number. For example, the list menu 1 is stored in the variable file named “MENU301.VR”.

Procedure 36-3 Setup List Menu

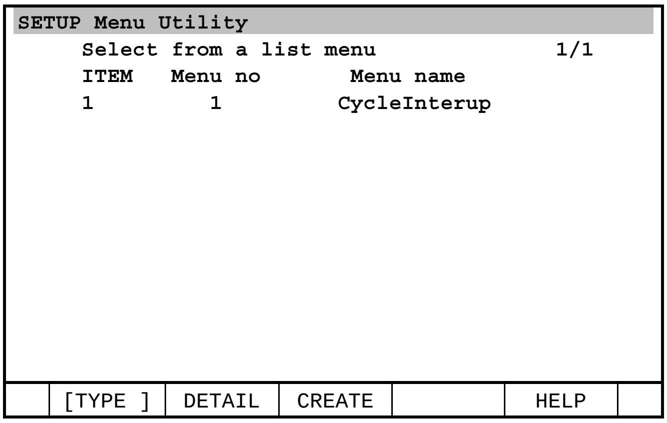
Step

1 Press [MENU] key and select “6 SETUP” and select ”Menu Utility” so that Menu utility screen will be displayed.



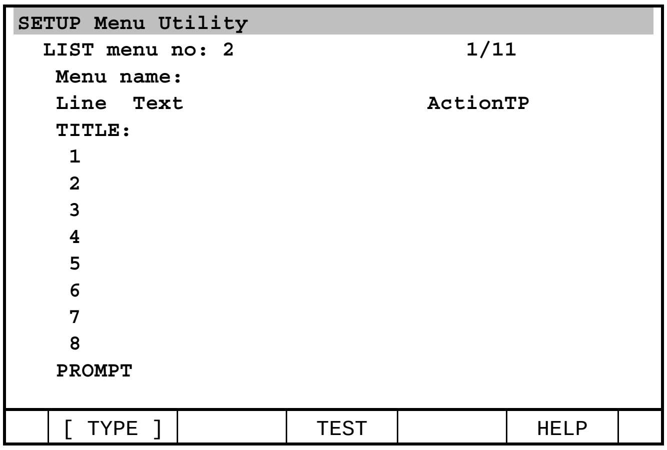
```
SETUP Menu Utility
  Choose a menu type:                       1/5

   1 Prompt box msg
   2 Prompt box yes/no
   3 Select from a list
   4 Status menus
   5 Operator entry
```

2 Set cursor to “3 Select from a list” and press F2, DETAIL so that Prompt box will be displayed. You will see a screen similar to the following.



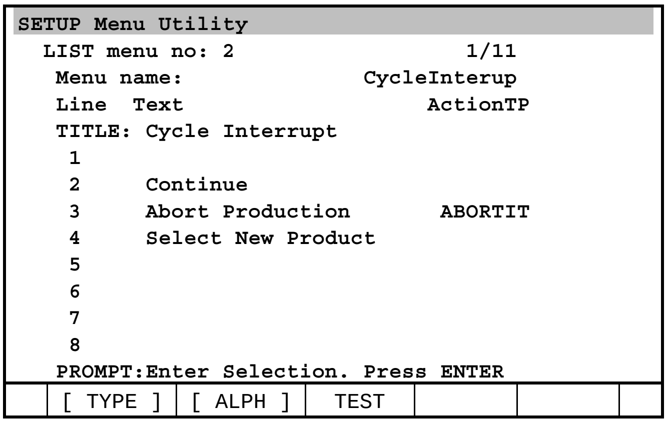
```
SETUP Menu Utility
     Select from a list menu               1/1
     ITEM   Menu no       Menu name
     1         1         CycleInterup
```

3 Press F3, CREATE if you want to create new menu. Press F2, DETAIL if you want to change menu that already exists. You will see a screen similar to the following.



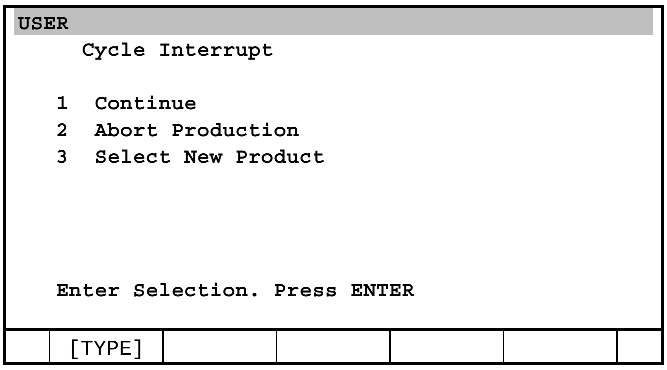
```
SETUP Menu Utility
  LIST menu no: 2                   1/11
   Menu name:
   Line  Text                  ActionTP
   TITLE:
    1
    2
    3
    4
    5
    6
    7
    8
   PROMPT
```

4 Input menu name to line “Menu name” and text to each line. Line that is input text becomes item when list menu is displayed. Set cursor to Action TP and press F4, CHOICE and select teach pendant program if you want to associate an Action TP program with a list item.



```
SETUP Menu Utility
  LIST menu no: 2                  1/11
   Menu name:              CycleInterup
   Line  Text                   ActionTP
   TITLE: Cycle Interrupt
    1
    2     Continue
    3     Abort Production       ABORTIT
    4     Select New Product
    5
    6
    7
    8
   PROMPT:Enter Selection. Press ENTER
```

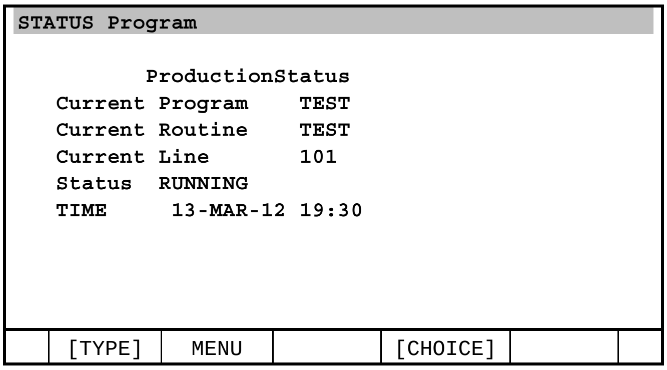
5 Press F3, TEST if you want to confirm the screen that will be displayed. You will see a screen similar to the following.



```
USER
     Cycle Interrupt

   1  Continue
   2  Abort Production
   3  Select New Product




   Enter Selection. Press ENTER
```

36.1.4 Status Menu

A status menu displays status information to the teach pendant screen. When a status menu is executed, the teach pendant displays the status menu and then continues executing. The values that are displayed are refreshed at least once every second. This is read-only menu.



```
STATUS Program

          ProductionStatus
   Current Program    TEST
   Current Routine    TEST
   Current Line       101
   Status  RUNNING
   TIME     13-MAR-12 19:30
```

Macro: Status Menu(menu)
A status menu uses the Status Menu(menu) macro.

Parameter 1: menu
This is the menu number or menu name. The menu can be chosen by number or name.

Usage example:
  Example 1:
    PROG TEST1

1: Status Menu (2)

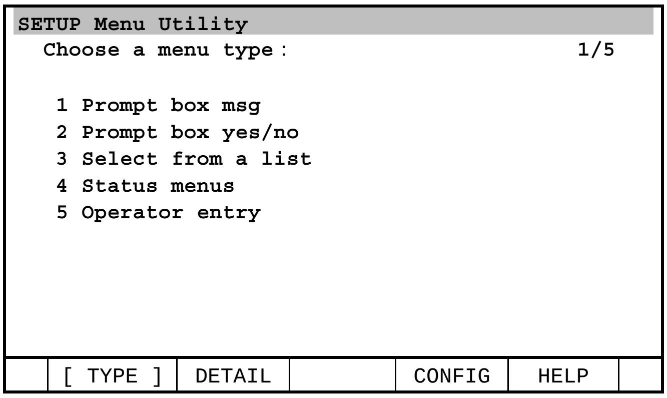
Example 2:
PROG TEST2
1: Status Menu (‘ProductionStatus’)

Save Menu setting:
Each list menu is stored in its own variable file that begins with the prefix “MENU4”. The next two characters of the filename are the menu number. For example, the list menu 1 is stored in the variable file named “MENU401.VR”.

Procedure 36-4 Setup Status Menu

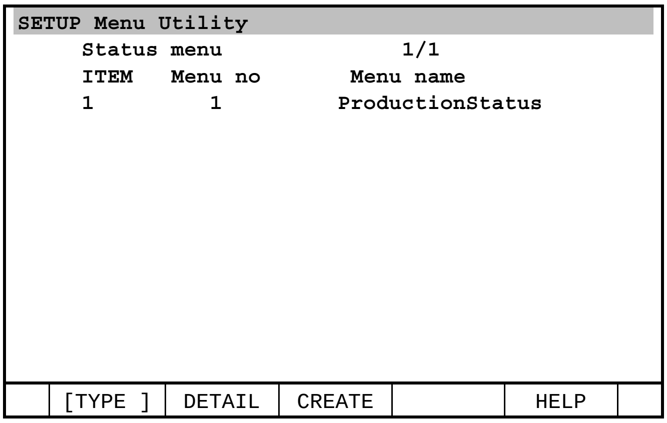
Step

1 Press [MENU] key and select ”6 SETUP” and select ”Menu Utility” so that Menu utility screen will be displayed .



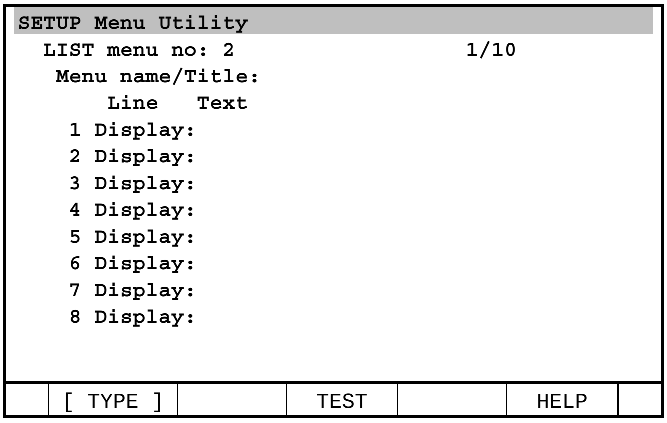
```
SETUP Menu Utility
  Choose a menu type :                         1/5

   1 Prompt box msg
   2 Prompt box yes/no
   3 Select from a list
   4 Status menus
   5 Operator entry
```

2 Set cursor to “4 Status menus” and press F2, DETAIL so that Prompt box will be displayed. You will see a screen similar to the following.



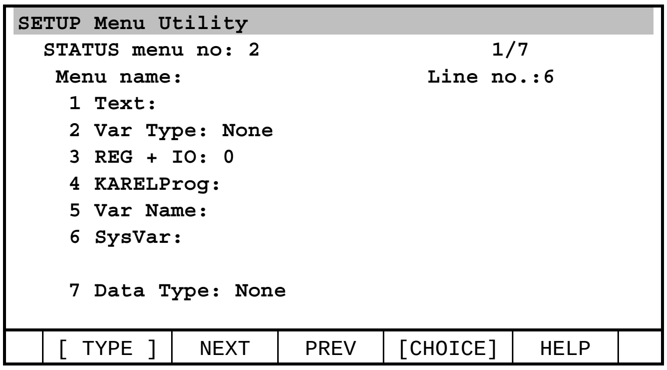
```
SETUP Menu Utility
     Status menu               1/1
     ITEM   Menu no        Menu name
     1          1         ProductionStatus
```

3 Press F3, CREATE if you want to create new menu. Press F2, DETAIL if you want to change menu that already exists. You will see a screen similar to the following.



```
SETUP Menu Utility
  LIST menu no: 2                  1/10
   Menu name/Title:
       Line   Text
    1 Display:
    2 Display:
    3 Display:
    4 Display:
    5 Display:
    6 Display:
    7 Display:
    8 Display:
```

4 Set cursor to line that you want to edit, and press F2, DETAIL. You will see a screen similar to the following.



```
SETUP Menu Utility
  STATUS menu no: 2                  1/7
   Menu name:                   Line no.:6
    1 Text:
    2 Var Type: None
    3 REG + IO: 0
    4 KARELProg:
    5 Var Name:
    6 SysVar:

    7 Data Type: None
```

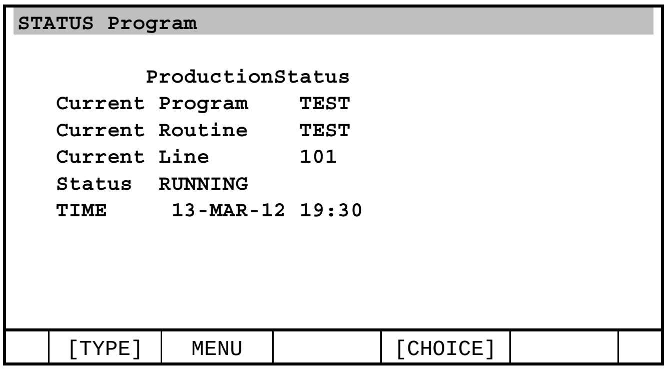
5 Setup each item. It is possible to setup if you set cursor objective item and press F4, [CHOICE]. Please refer to following table and setup each item.

Table 36.1.4 Status menu variable detail information

ITEM	DESCRIPTION		
Text	This item allows you to specify the text that will be displayed for the status item. You can type up to 17 characters of display text.		
Var Type	This item allows you to specify the kind of variable for the status item:		
	Type of Variable	**What to Define**	**Data Display Type**
	NONE	None	none
	R[]	REG Number	INTEGER, REAL, BOOLEAN
	DIN[]	I/O Number	INTEGER, BOOLEAN, TEXT
	DOUT[]	I/O Number	INTEGER, BOOLEAN, TEXT
	RI[]	I/O Number	INTEGER, BOOLEAN, TEXT
	RO[]	I/O Number	INTEGER, BOOLEAN, TEXT
	GIN[]	I/O Number	INTEGER
	GOUT[]	I/O Number	INTEGER
	KAREL VAR	Prog. Name, Var. Name	INTEGER, REAL, BOOLEAN,TEXT
	SYSTEM VAR	System Variable Name	INTEGER, REAL, BOOLEAN,TEXT
	TIME	None	----
	CURR PROGRAM	None	----
	CURR ROUTINE	None	----
	TIMER (SEC)	Prog. Name, Var. Name	----
	TIMER (MIN)	Prog. Name, Var. Name	----
	TIMER (HR)	Prog. Name, Var. Name	----
	CURR LINE	None	----

ITEM	DESCRIPTION
REG＋IO	This item allows you to specify the number of the register or the I/O signal, when the variable type is R[ ],DIN[ ],DOUT[ ],RI[ ],RO[ ],GIN[ ], or GOUT[ ].
KAREL Program	This item allows you to specify the name of the KAREL program that contains the variable you are displaying, when the variable type is KAREL VAR.
Variable Name	This item allows you to specify the name of the variable you are displaying.
System Variable	This item allows you to specify the name of the system variable you are displaying, when the variable type is SYSTEM VAR.
Data Type	This item allows you to specify the variable data types: ・ NONE ・ INTEGER ・ REAL ・ TEXT ・ BOOLEAN

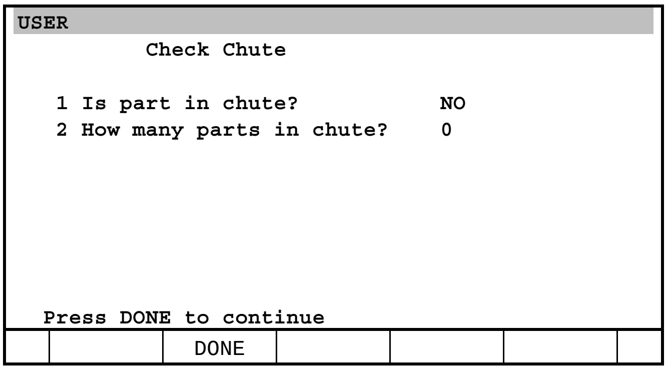
6 Press F3, TEST if you want to confirm the screen that will be displayed in the step 3. You will see a screen similar to the following.



```
STATUS Program

          ProductionStatus
   Current Program    TEST
   Current Routine    TEST
   Current Line       101
   Status  RUNNING
   TIME     13-MAR-12 19:30
```

36.1.5 Operator Entry Menu

An operator entry menu requires the operator to enter INTEGER, REAL, BOOLEAN, or text values. When an operator entry menu is executed, the program pauses until the operator types the information and then presses F2, DONE.



```
USER
          Check Chute

   1 Is part in chute?           NO
   2 How many parts in chute?    0







  Press DONE to continue
```

Macro: Op. Entry Menu(menu)
An operator entry menu uses the Op. Entry Menu(menu) macro.

Parameter 1: menu
This is the menu number or menu name. The menu can be chosen by number or name.

Usage example:

Example 1:

```
PROG   TEST1
    1: Status Menu (2)
```

Example 2:

```
PROG   TEST2
    1: Status Menu ('ProductionStatus')
```

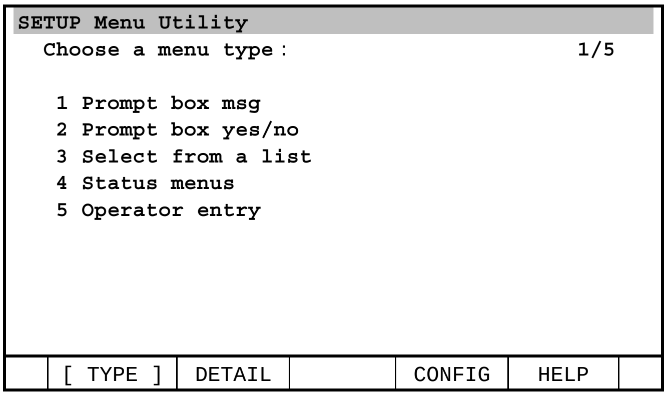
Save Menu setting:

Each operator entry menu is stored in its own variable file that begins with the prefix “MENU5”. The next two characters of the filename are the menu number. For example, the list menu 1 is stored in the variable file named “MENU501.VR”.

Procedure 36-5 Setup Operator entry

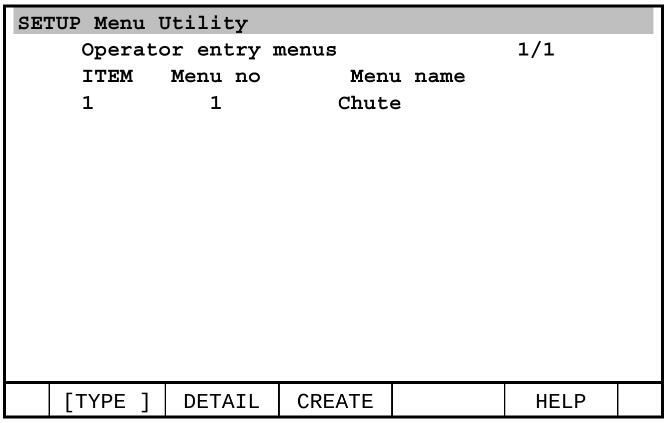
Step

1 Press [MENU] key and select “6 SETUP” and select ”Menu Utility” so that Menu utility screen will be displayed .



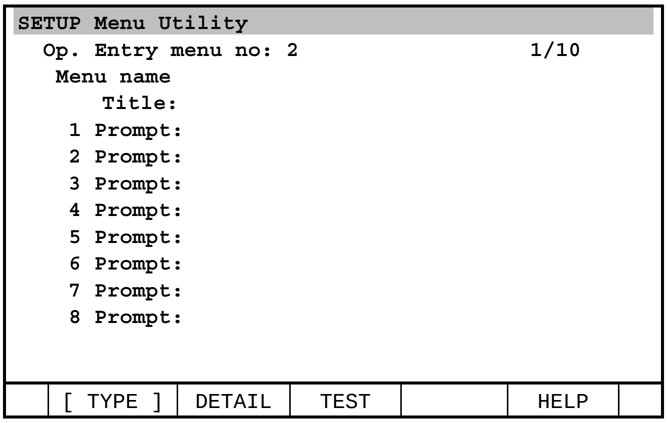
```
SETUP Menu Utility
  Choose a menu type :                        1/5

   1 Prompt box msg
   2 Prompt box yes/no
   3 Select from a list
   4 Status menus
   5 Operator entry

   [ TYPE ]  DETAIL           CONFIG    HELP
```

2 Set cursor to “5 Operator entry” and press F2, DETAIL so that Prompt box will be displayed. You will see a screen similar to the following.



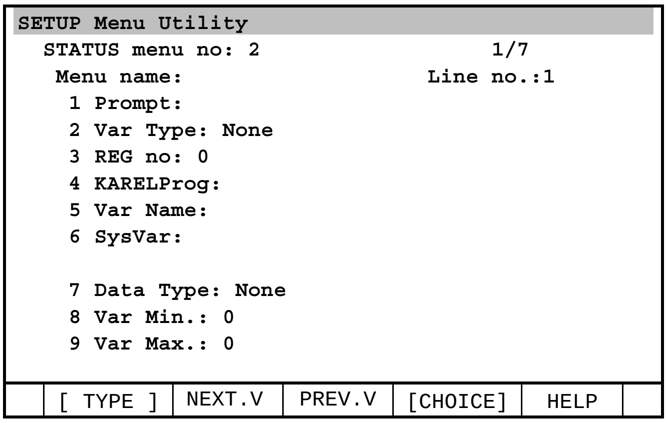
```
SETUP Menu Utility
     Operator entry menus                1/1
     ITEM   Menu no       Menu name
     1         1         Chute

   [TYPE ]  DETAIL   CREATE             HELP
```

3 Press F3, CREATE if you want to create new menu. Press F2, DETAIL if you want to change menu that already exists. You will see a screen similar to the following.



```
SETUP Menu Utility
  Op. Entry menu no: 2                  1/10
   Menu name
      Title:
    1 Prompt:
    2 Prompt:
    3 Prompt:
    4 Prompt:
    5 Prompt:
    6 Prompt:
    7 Prompt:
    8 Prompt:

   [ TYPE ]  DETAIL   TEST          HELP
```

4 Set cursor to line that you want to edit, and press F2, DETAIL. The following screen will be displayed.



```
SETUP Menu Utility
  STATUS menu no: 2                  1/7
   Menu name:                  Line no.:1
    1 Prompt:
    2 Var Type: None
    3 REG no: 0
    4 KARELProg:
    5 Var Name:
    6 SysVar:

    7 Data Type: None
    8 Var Min.: 0
    9 Var Max.: 0

  [ TYPE ]  NEXT.V  PREV.V  [CHOICE]  HELP
```

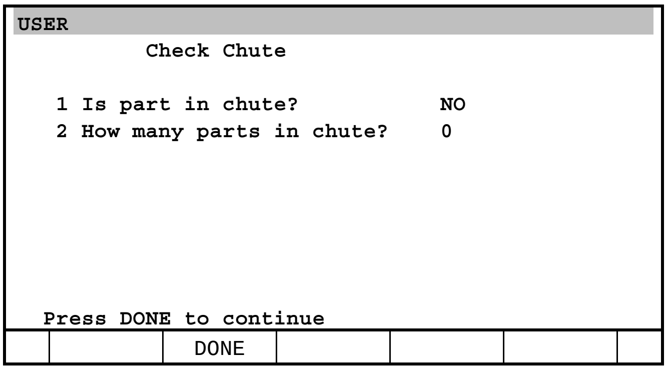
5 Setup each item. It is possible to setup if you set cursor objective item and press F4, [CHOICE] or [ENTER] key. Please refer to following table and setup each item.

Table 36.1.5 Operator entry menu variable detail information

<table>
<tr><th>ITEM</th><th colspan="3">DESCRIPTION</th></tr>
<tr><td><b>Prompt</b></td><td colspan="3">This item allows you to specify the text that will be displayed for the prompt item. You can type up to 17 characters of display text.</td></tr>
<tr><td rowspan="6"><b>Variable</b></td><td colspan="3">This item allows you to specify the kind of variable for the value of the prompt item.</td></tr>
<tr><td><b>Type of Variable</b></td><td><b>What to Define</b></td><td><b>Data Display Type</b></td></tr>
<tr><td>NONE</td><td>None</td><td>None</td></tr>
<tr><td>R[]</td><td>REG Number</td><td>INTEGER, REAL, BOOLEAN</td></tr>
<tr><td>KAREL VAR</td><td>Prog.Name, Var.Name</td><td>INTEGER, REAL, BOOLEAN, TEXT</td></tr>
<tr><td>SYSTEM VAR</td><td>System Variable</td><td>INTEGER, REAL, BOOLEAN, TEXT</td></tr>
<tr><td><b>REG Number</b></td><td colspan="3">This item allows you to specify the number of the register, When the variable type is R[].</td></tr>
<tr><td><b>KAREL Program</b></td><td colspan="3">This item allows you to specify the name of the KAREL program that contains the variable you are using, when the variable type is KAREL VAR.</td></tr>
<tr><td><b>Variable Name</b></td><td colspan="3">This item allows you to specify the name of the KAREL variable you are using, when the variable type is KAREL VAR.</td></tr>
<tr><td><b>System Variable</b></td><td colspan="3">This item allows you to specify the name of the system variable you are using, when the variable type is SYSTEM VAR.</td></tr>
</table>

ITEM	DESCRIPTION
Data Type	This item allows you to specify the variable data types: ・ NONE ・ INTEGER ・ REAL ・ TEXT ・ BOOLEAN
Variable Minimum	This item allows you to specify the minimum value allowed for the INTEGER or REAL variables.
Variable Maximum	This item allows you to specify the maximum value allowed for the INTEGER or REAL variables.

6 Press F3, TEST if you want to confirm the screen that will be displayed in the step 3. The following screen will be displayed.



```
USER
          Check Chute

   1 Is part in chute?           NO
   2 How many parts in chute?    0







  Press DONE to continue
```

37 4D GRAPHICS FUNCTION

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 4th dimensional information.
4D means the fusion of 3D robot model and 1D internal data as 4th 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.

To use this function, 4D GRAPHICS option (R764) is required.

> ⚠ **WARNING**
> When moving the robot with the pendant enabled, be sure to watch the robot instead of watching the TP 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.

The invisible data that can be displayed are:

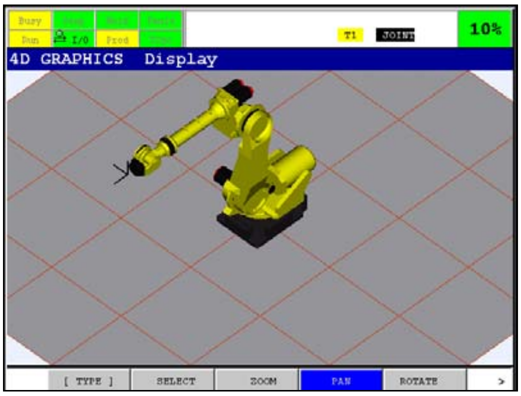
1	Logical tool center point (TCP).	- 4D graphics display function
2	Frame settings	- 4D visual jog function
3	Robot jog settings	- 4D jog preview function
4	The points (node) in a TP program	- 4D node map function
5	Exact path that the robot followed	- 4D TCP trace function
6	Position registers	- 4D position register function
7	DCS safety zone (*1)	- 4D DCS function.

*1: This function requires DCS Pos./Seed check (J567) option.

Procedure 37-1 4D GRAPHICS Display screen

Step

1 Press [MENU] key to display the screen menu.
2 Press “0 NEXT,” then select “5 4D GRAPHICS”. The following screen will be displayed.
3 Alternatively, instead of steps 1 and 2 above, the user can press *i* key and POSN key at the same time.



37.1 OVERVIEW

37.1.1 Graphic Models

4D GRAPHICS screen displays the following graphic models.

- Robot model
- Internal data
- Cell floor
- Peripheral devices

Robot model

A graphic model of ordered robot is displayed.

Internal data

Various data such as positions taught in a program or tool center point are displayed.

Cell floor

A reference floor with a 1 meter grid. The position of cell floor can be set in SETUP Frames screen.

Peripheral devices

Peripheral device graphic models such as a works, fences and tools can be located anywhere in 4D GRAPHICS scene by using 4D Editor function in ROBOGUIDE. Please refer to section 37.3 4D Editor Function for details.

37.1.2 Operation Procedure

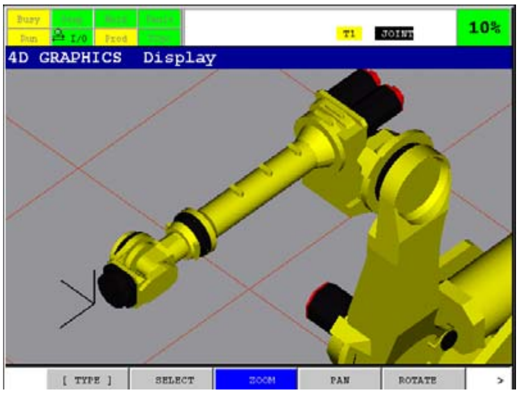
In 4D GRAPHICS screen, view is adjusted by the following operation modes.

- 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 iPendant key input and the other is touch panel operation.

Zoom by iPendant key input

Increase the magnification

- Press the up arrow key on iPendant.
- Press the SHIFT + up arrow key on iPendant. (High magnification)

Decrease the magnification

- Press the down arrow key on iPendant.
- Press the SHIFT + down arrow key on iPendant. (High magnification)

Zoom by touch panel

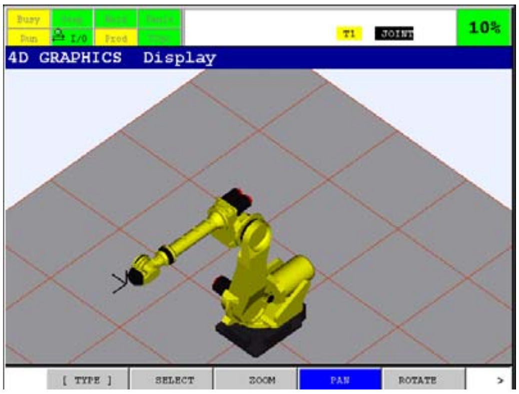
Increase the magnification

- 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 iPendant key input and the other is touch panel operation.

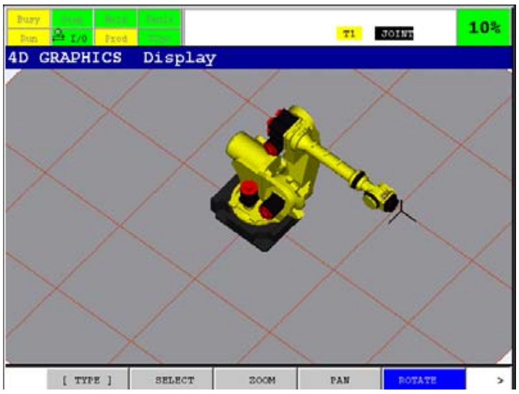
PAN by iPendant key input

- 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 iPendant key input and the other is touch panel operation.

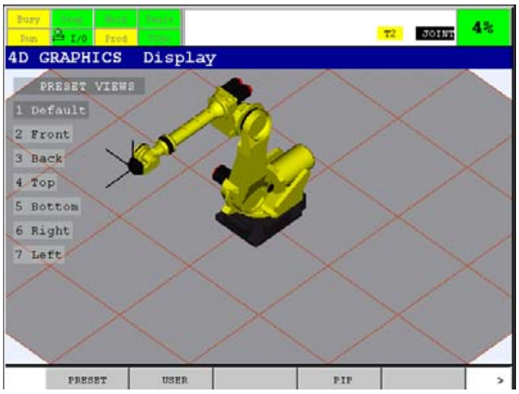
ROTATE by iPendant key input

- Press the arrow key on iPendant up, down, left and right.
- Press the SHIFT + arrow key on iPendant 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
- Left Set the view on the left side of the robot
- Right Set the view on the right side of the robot

To select Preset Views, press NEXT key and press F2, [VIEWS]. Select one of the preset views above.

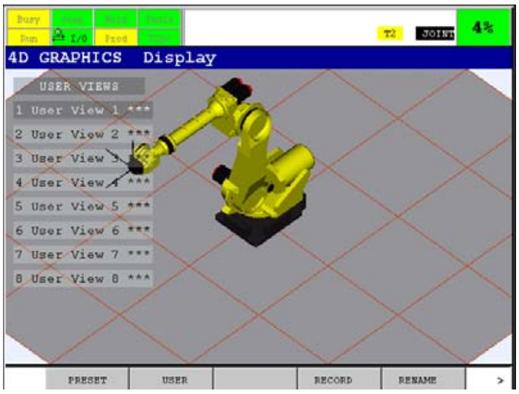
User Views

4D GRAPHICS screen provides eight user views. These views can be recorded and then retrieved by selecting them.

Procedure 37-2 Record User Views

Step

1 Move to the view you want to record.
2 Press [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.

37.2 4D GRAPHICS SCENE

In 4D GRAPHICS function, the following scenes are available.

- 4D GRAPHICS Display
- 4D GRAPHICS Edit Node Map
- 4D GRAPHICS Select Node Map
- 4D GRAPHICS Position Register
- 4D GRAPHICS TCP Trace
- 4D GRAPHICS Frame Display
- 4D GRAPHICS CDS (This function requires DCS Pos./Seed check (J567) option.)

Pressing F1, [TYPE] in any 4D GRAPHIC scene and selecting one of them display the selected scene.

37.2.1 4D GRAPHICS Display

4D GRAPHICS Display scene is the standard scene and it can composite together any multiple scenes to be viewed together.

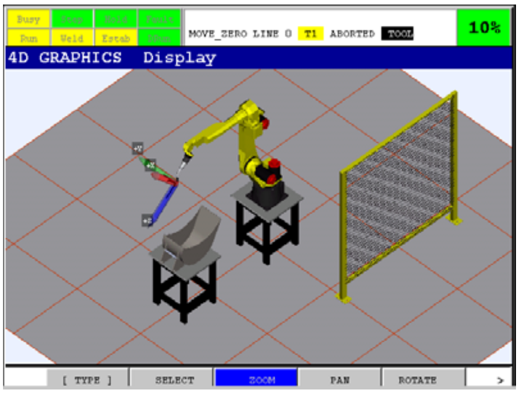
In 4D GRAPHICS Display scene, the following functions are available.

- Visual Jog
- Jog Preview

37.2.1.1 Visual jog

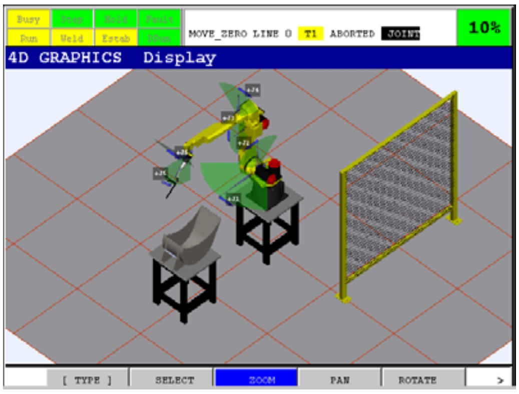
Visual Jog allows the coordinate system and the group selected for jogging to be indicated on the 4D GRAPHICS Display. It allows you to preview the direction that the robot will move before actually jogging the robot.
Visual Jog is available when the teach pendant is enable.

Cartesian Visual Jog



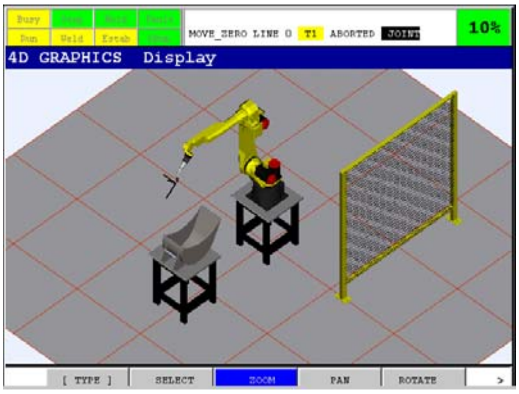
The red, green and blue axes correspond to the X, Y and Z directions.

Joint Visual Jog



Each joint has an indicator consisting of a semi-transparent fan with a needle showing the current position. The fan is usually green and turns red as the robot is jogged closer to the limit.

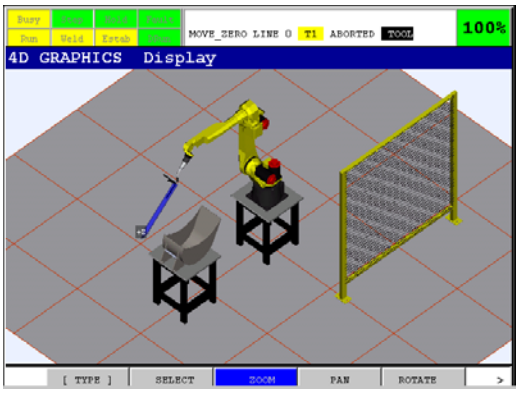
Pressing *i* key and COORD key simultaneously turn this function on and off.



37.2.1.2 Jog preview

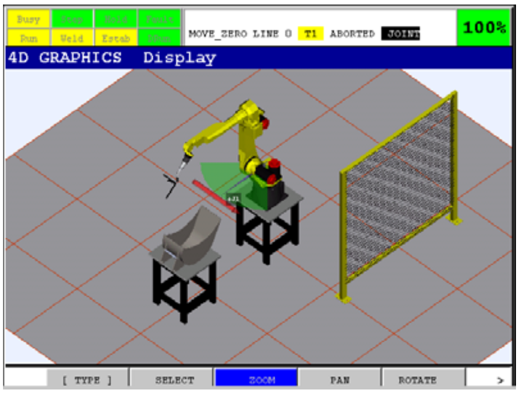
Jog Preview can preview the direction of movement of the robot when a particular jog key is pressed. It is activated when pressing a single jog key at the same time as the *i* key. Pressing more than one jog key will display the jog direction of the last key pressed. It is not necessary to enable the DEADMAN or clean errors. Jog Preview is available when the teach pendant is enable.

Cartesian Jog Preview



Cartesian Jog Preview shows a single arrow indicating the direction that the robot will move. Depending on the setting of the speed override, the size of the arrow will change. The figure shows an example when *i* key and +Z jog key.

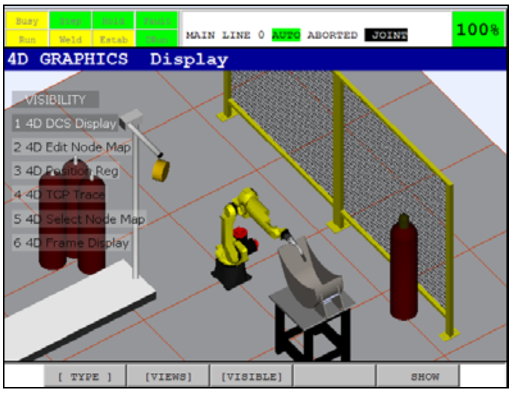
Joint Jog Preview



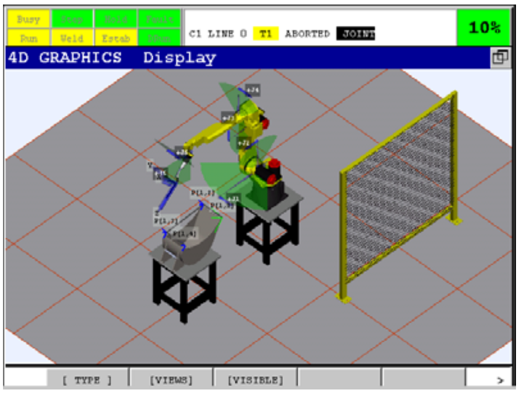
Joint Jog Preview shows a single red arrow tangent to the fan indicating the direction that the joint will move. Depending on the setting of the speed override, the size of the arrow will change. The figure shows an example when *i* key and +J1 jog key.

37.2.1.3 Set visibility: 4D GRAPHICS display

The elements which are visible in 4D GRAPHICS Display scene is controlled by the visibility softkey on page two of the softkeys. Pressing F3, [VISIBLE] on page two of the softkeys shows the following menu.



For example, selecting 4D Edit Node Map adds the node map in the scene. Selecting the same item again to make the element invisible. In the following figure, the node map and frame elements are added.



> **NOTE**
> When there are too much information displayed, the performance of 4D GRAPHICS function declines.
> In that case, make unnecessary data invisible.

37.2.2 4D GRAPHICS Node Map

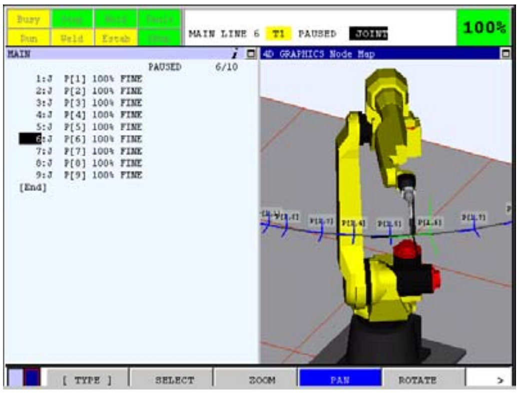
Node map is a 4D visualization of a TP program. This shows all the positions as nodes and connect them with lines. Each node has its position number in the text label elements. The main purpose of node map is to see the relationship among the positions in a program without moving a robot.

> **NOTE**
> Node Map connects two nodes in the program in descending order. Therefore, it is sometimes different from the real motion path.

> **NOTE**
> When INC, OFFSET or Position Register is used in a motion statement or when frame number is changed in a program, the node will be displayed in the different position from the actual position.

Each node is colored in blue, green or yellow. The blue node means normal, the green one means the position focused in EDIT screen and the yellow one means the position is unspecified, such as the position register. The line between nodes is usually colored in gray or red. The red one means there is an unspecified position between the two nodes.
Edit node map and select node map are functionally a little different from each other.

37.2.2.1 4D GRAPHICS edit node map



```
                              PAUSED    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]
```

In 4D GRAPHICS Edit Node Map scene, the currently selected program is displayed. It is linked to EDIT screen. So the focused position is displayed as a green node in node map. As the cursor is moved in EDIT screen, the green node also changes. The node map will be updated every time the program is edited.

Procedure 37-3 4D GRAPHICS Edit Node Map

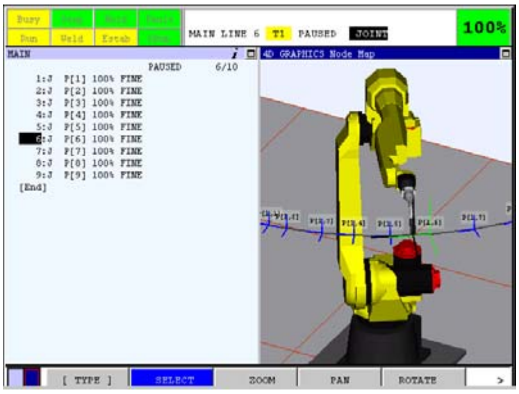
Step

There are some other ways to display 4D GRAPHICS Edit Node Map in addition to selecting the item from F1, [TYPE].

1 Press [EDIT] key. When a program is selected, EDIT screen will be displayed.
2 Press [*i*] key and [FCTN] key on EDIT screen. Related View menu will be displayed.
3 Select "4D Edit Node Map".
4 Instead of step 1 to step 3, Pressing *i* key and EDIT key displays the node map.

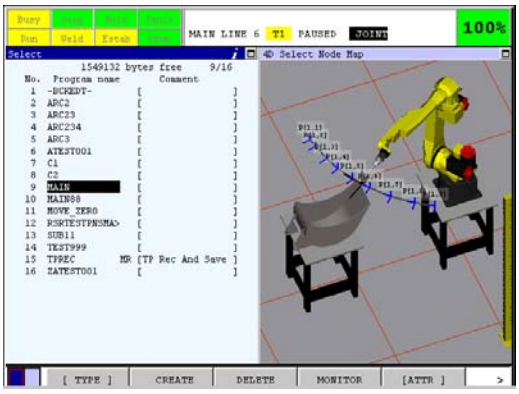
SELECT Data

Selecting a node can make the cursor in EDIT screen move to the corresponding line. To select a node, touch the screen near the node. To use this functionality, select F2, SELECT. Then, the label F2, SELECT turns blue in color.



```
MAIN LINE 6  T1  PAUSED  JOINT        100%
MAIN
                         PAUSED     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]
```

37.2.2.2 4D GRAPHICS select node map



```
MAIN LINE 6  T1  PAUSED  JOINT        100%
Select
          1549132 bytes free      9/16
 No.  Program name       Comment
  1  -BCKEDT-         [               ]
  2  ARC2             [               ]
  3  ARC23            [               ]
  4  ARC234           [               ]
  5  ARC3             [               ]
  6  ATEST001         [               ]
  7  C1               [               ]
  8  C2               [               ]
  9  MAIN             [               ]
 10  MAIN88           [               ]
 11  MOVE_ZERO        [               ]
 12  RSRTESTPNSMA>    [               ]
 13  SUB11            [               ]
 14  TEST999          [               ]
 15  TPREC         MR [TP Rec And Save ]
 16  ZATEST001        [               ]
```

In 4D GRAPHICS Select Node Map scene, the currently focused program in SELECT screen is displayed. It is linked to SELECT screen. As the cursor is moved in SELECT screen, displayed node map also changes. When a program is selected in SELECT screen, EDIT screen of the program is displayed. Then, 4D GRAPHICS Edit Node Map is also displayed automatically.

Procedure 37-4 4D GRAPHICS Select Node Map

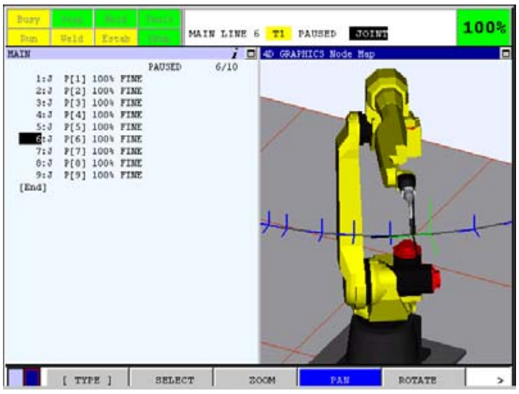
Step

There are some other ways to display 4D GRAPHICS Select Node Map in addition to selecting the item from F1, [TYPE].

1 Press [SELECT] key. SELECT screen will be displayed.
2 Press [*i*] key and [FCTN] key on SELECT screen. Related View menu will be displayed.
3 Select “4D Display”.
4 Instead of step 1 to step 3, Pressing [*i*] key and [SELECT] key displays the node map.

37.2.2.3 Set Visibility: node map position number

The position number element can be turned on and off. To do it, press NEXT and then press F3, [VISIBLE]. The following figure shows a node map without the position number.



```
MAIN LINE 6  T1  PAUSED  JOINT
MAIN
                              PAUSED      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]
```

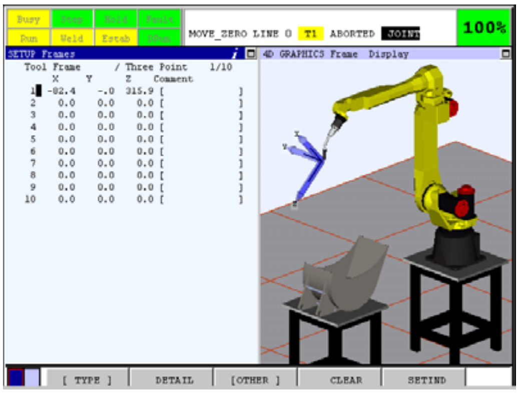
37.2.2.4 Data supported by node map

The following data are supported in node map.

- Normal positions
- Positions specified by Position Register
- Via positions and end positions in Circular motion
- Incremented position by INC option

37.2.3 4D GRAPHICS Frame Display

4D GRAPHICS Frame Display scene shows the following graphics of the current frame setup operation. The comments for the frame is also displayed.



```
MOVE_ZERO LINE 0  T1  ABORTED  JOINT
SETUP Frames
  Tool Frame         / Three Point    1/10
        X      Y       Z    Comment
  1  -82.4    -.0   315.9 [                 ]
  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 [                 ]
```

Procedure 37-5 4D GRAPHICS Frame Display

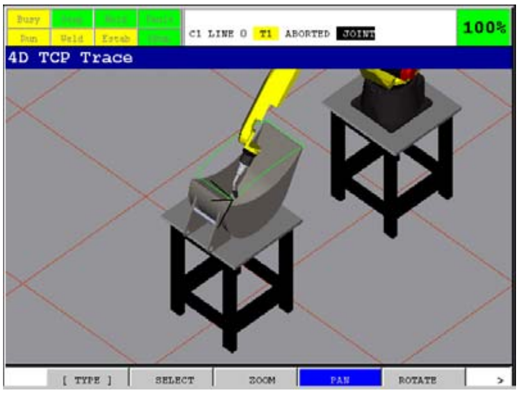
Step

There are another way to display 4D GRAPHICS Frame Display in addition to selecting the item from F1, [TYPE].

1 Press [MENU] key.
2 Select “6 Setup”.
3 Press F1, [TYPE].
4 Select “Frames”.
5 Press [*i*] key and [FCTN] key on Frame setup screen. Related View menu will be displayed.
6 Select “4D Frame Display”.

37.2.4 4D GRAPHICS TCP Trace

4D GRAPHICS TCP Trace scene provides the capability to view the path that the robot took when executing a program. The path is displayed in green. The path is updated every time a program is executed.

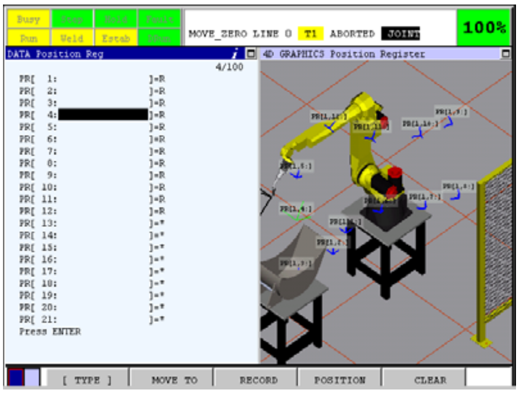


> **NOTE**
> The path provided in 4D GRAPHICS TCP Trace are possibly different from actual path.

37.2.5 4D GRAPHICS Position Register

In 4D GRAPHICS Position Register scene, position registers are displayed as nodes like node map. This helps understand where the position registers are located spatially. Each node has its position register number.

4D GRAPHICS Position Register is linked to Position Register screen. Focused position register is displayed as green highlighted node. When a position register is newly taught or modified, the graphics are automatically updated.



```
                                        4/100
  PR[  1:                      ]=R
  PR[  2:                      ]=R
  PR[  3:                      ]=R
  PR[  4:                      ]=R
  PR[  5:                      ]=R
  PR[  6:                      ]=R
  PR[  7:                      ]=R
  PR[  8:                      ]=R
  PR[  9:                      ]=R
  PR[ 10:                      ]=R
  PR[ 11:                      ]=R
  PR[ 12:                      ]=R
  PR[ 13:                      ]=*
  PR[ 14:                      ]=*
  PR[ 15:                      ]=*
  PR[ 16:                      ]=*
  PR[ 17:                      ]=*
  PR[ 18:                      ]=*
  PR[ 19:                      ]=*
  PR[ 20:                      ]=*
  PR[ 21:                      ]=*
  Press ENTER
```

Procedure 37-6 4D GRAPHICS Position Register

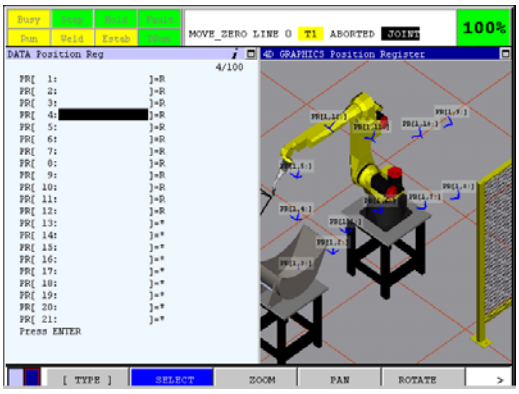
Step

There are another way to display 4D GRAPHICS Frame Display in addition to selecting the item from F1, [TYPE].

1 Press [DATA] key and display DATA screen.
2 Press F1, [TYPE].
3 Select "Position Reg".
4 Press [*i*] key and [FCTN] key on Position Register screen. Related View menu will be displayed.
5 Select "4D Position Reg".
6 Instead of step 1 to step 5, pressing *i* key and DATA key display the screen.

SELECT Data

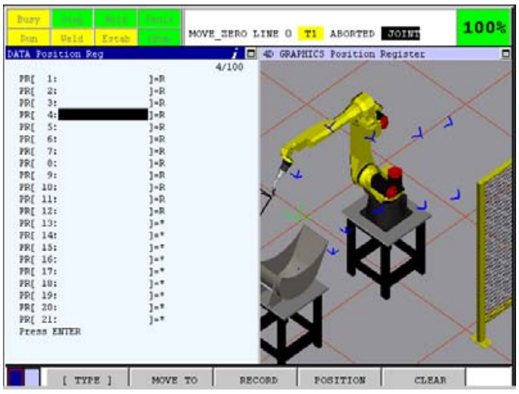
Selecting a node can make the cursor in Position Register screen move to the corresponding line. To select a node, touch the screen near the node. To use this functionality, select F2, SELECT. Then, the label F2, SELECT turns blue in color.



```
4/100
PR[  1:          ]=R
PR[  2:          ]=R
PR[  3:          ]=R
PR[  4:          ]=R
PR[  5:          ]=R
PR[  6:          ]=R
PR[  7:          ]=R
PR[  8:          ]=R
PR[  9:          ]=R
PR[ 10:          ]=R
PR[ 11:          ]=R
PR[ 12:          ]=R
PR[ 13:          ]=*
PR[ 14:          ]=*
PR[ 15:          ]=*
PR[ 16:          ]=*
PR[ 17:          ]=*
PR[ 18:          ]=*
PR[ 19:          ]=*
PR[ 20:          ]=*
PR[ 21:          ]=*
Press ENTER
```

37.2.5.1 Set Visibility: position register number

The position register number element can be turned on and off. To do it, press NEXT and then press F3, [VISIBLE]. The following figure shows a node map without the position register number.



```
4/100
PR[  1:          ]=R
PR[  2:          ]=R
PR[  3:          ]=R
PR[  4:          ]=R
PR[  5:          ]=R
PR[  6:          ]=R
PR[  7:          ]=R
PR[  8:          ]=R
PR[  9:          ]=R
PR[ 10:          ]=R
PR[ 11:          ]=R
PR[ 12:          ]=R
PR[ 13:          ]=*
PR[ 14:          ]=*
PR[ 15:          ]=*
PR[ 16:          ]=*
PR[ 17:          ]=*
PR[ 18:          ]=*
PR[ 19:          ]=*
PR[ 20:          ]=*
PR[ 21:          ]=*
Press ENTER
```

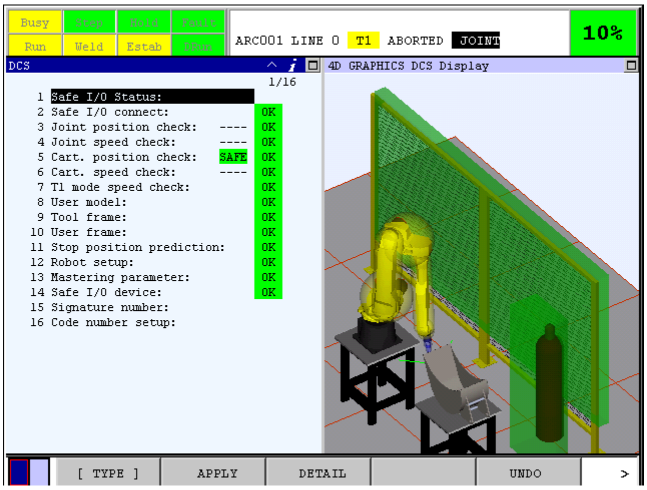
NOTE

Registers are always considered to be in the "current" frame. The register graphical display reflects the position of the register in the current frame. This may not be how the register is used. Changing the current frame will be reflected in the position of the register triad in the graphics display.

> **NOTE**
> Any register which has been set to joint representation will not be displayed.

37.2.6 4D GRAPHICS DCS

In 4D GRAPHICS DCS scene, DCS safety zones are displayed. By this function DCS safety zones can be set and check graphically. This section describes the way of displaying 4D DCS screen.
The robot controller must have the DCS Pos./Speed check (J567) option installed to use this function.
Please refer to section 4 DCS VISUALIZATION in R-30*i*B/R-30*i*B Mate CONTROLLER Dual Check Safety Function OPERATOR'S MANUAL (B-83184EN) for details of 4D graphics DCS function.



```
ARC001 LINE 0  T1  ABORTED  JOINT

DCS                                   4D GRAPHICS DCS Display
                             1/16
  1 Safe I/O Status:
  2 Safe I/O connect:              OK
  3 Joint position check:    ----  OK
  4 Joint speed check:       ----  OK
  5 Cart. position check:    SAFE  OK
  6 Cart. speed check:       ----  OK
  7 T1 mode speed check:           OK
  8 User model:                    OK
  9 Tool frame:                    OK
 10 User frame:                    OK
 11 Stop position prediction:      OK
 12 Robot setup:                   OK
 13 Mastering parameter:           OK
 14 Safe I/O device:               OK
 15 Signature number:
 16 Code number setup:
```

Procedure 37-7 4D GRAPHICS DCS

Step

There is another way to display 4D GRAPHICS DCS in addition to selecting the item from F1 [TYPE] in 4D graphics display screen.

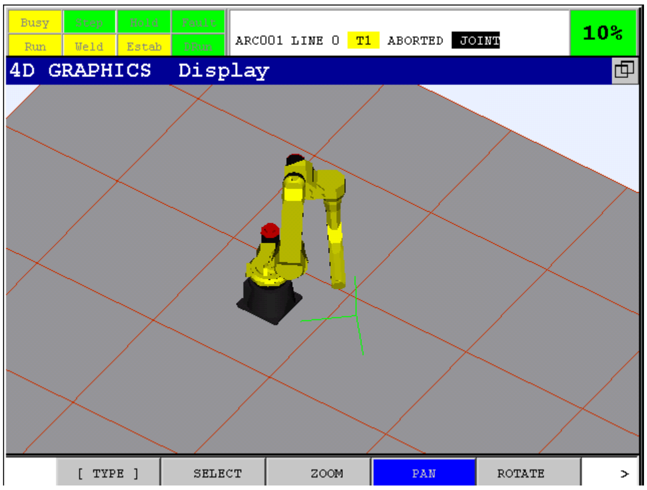
1 Press [MENU] key and display MENU pop up screen.
2 Select [0 -- NEXT --] and select [6 SYSTEM].
3 Press F1, [TYPE] and select [DCS].
4 Press [*i*] key and [FCTN] key on DCS screen. Related View menu will be displayed.
5 Select "4D DCS Display".

37.3 4D EDITOR FUNCTION

37.3.1 OVERVIEW

Peripheral device graphic models such as a works, fences and tools can be located anywhere in 4D GRAPHICS scene by using 4D Editor function in ROBOGUIDE. This section describes the screen using 4D Editor function.
Please refer to the ROBOGUIDE helps of 4D Editor for details of 4D Editor function.

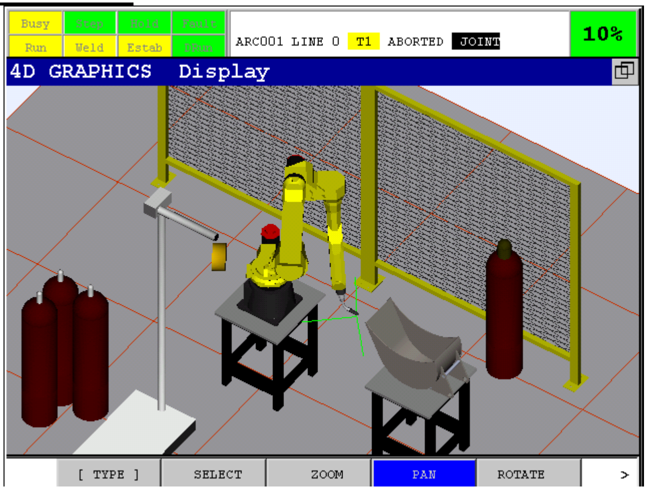
Graphics not using 4D Editor function



ARC001 LINE 0 T1 ABORTED JOINT 10%

4D GRAPHICS Display

Graphics using 4D Editor function



ARC001 LINE 0 T1 ABORTED JOINT 10%

4D GRAPHICS Display

4D Editor function enabled to display graphics more closely to the real system.

37.4 FULL SCREEN 4D DISPLAY

37.4.1 OVERVIEW

By this function, 4D graphics screen can be displayed on a PC.
This function requires the Internet Conn/Cust (R558) and R764 4D GRAPHICS function and (R764) option.

NOTE

For early releases, display of 4D information requires that the PC have an advanced graphics card capable of displaying OpenGL. If your PC does not work, updating your PC graphics drivers might fix it.

37.4.2 Setup

37.4.2.1 Requirements

The following are the requirements for Full Screen 4D Display.

- PC must have Microsoft® Internet Explorer 5.5 or greater installed.
- PC must have the *i*Pendant Controls installed. Please refer to section 15.2 iPENDANT CONTROLS INSTALLATION in R-30*i*A/R-30*i*A Mate/R-30*i*B CONTROLLER Ethernet Function OPERATOR'S MANUAL (B-82974EN) for installation instructions.
- PC must have an advanced graphics card capable of displaying OpenGL.
- PC must be connected to a network, and be properly configured to allow a TCP/IP connection to the Robot Controller with the *i*Pendant connected.
- The robot controller must be connected to a network and be properly configured for Network access to the above PC.
- The robot controller must have the Internet Conn/Custo (R558) option installed.
- The robot controller must have the 4D Graphics (R764) option installed. And the robot controller must have the DCS Pos./Speed check (J567) option installed to display 4D DCS Display.
- The HTTP Authentication must be properly configured on the robot to allow *i*Pendant access. Please refer to section 15.2 HTTP AUTHENTICATION in R-30*i*A/R-30*i*A Mate/R-30*i*B CONTROLLER Ethernet Function OPERATOR'S MANUAL (B-82974EN) for configuration information.

37.4.2.2 Configuring Microsoft® Internet Explorer

Please refer to section 15.2 HTTP AUTHENTICATION in R-30*i*A/R-30*i*A Mate/R-30*i*B /R-30*i*B Mate CONTROLLER Ethernet Function OPERATOR'S MANUAL (B-82974EN) for configuration information.

37.4.3 Operation

After you have properly configured Internet Explorer ® and verified that you can connect to the robot, you can now display the Full Screen 4D Display screen. The following sections detail the operation and the limitations of this feature.

37.4.3.1 Accessing The Full Screen 4D Display

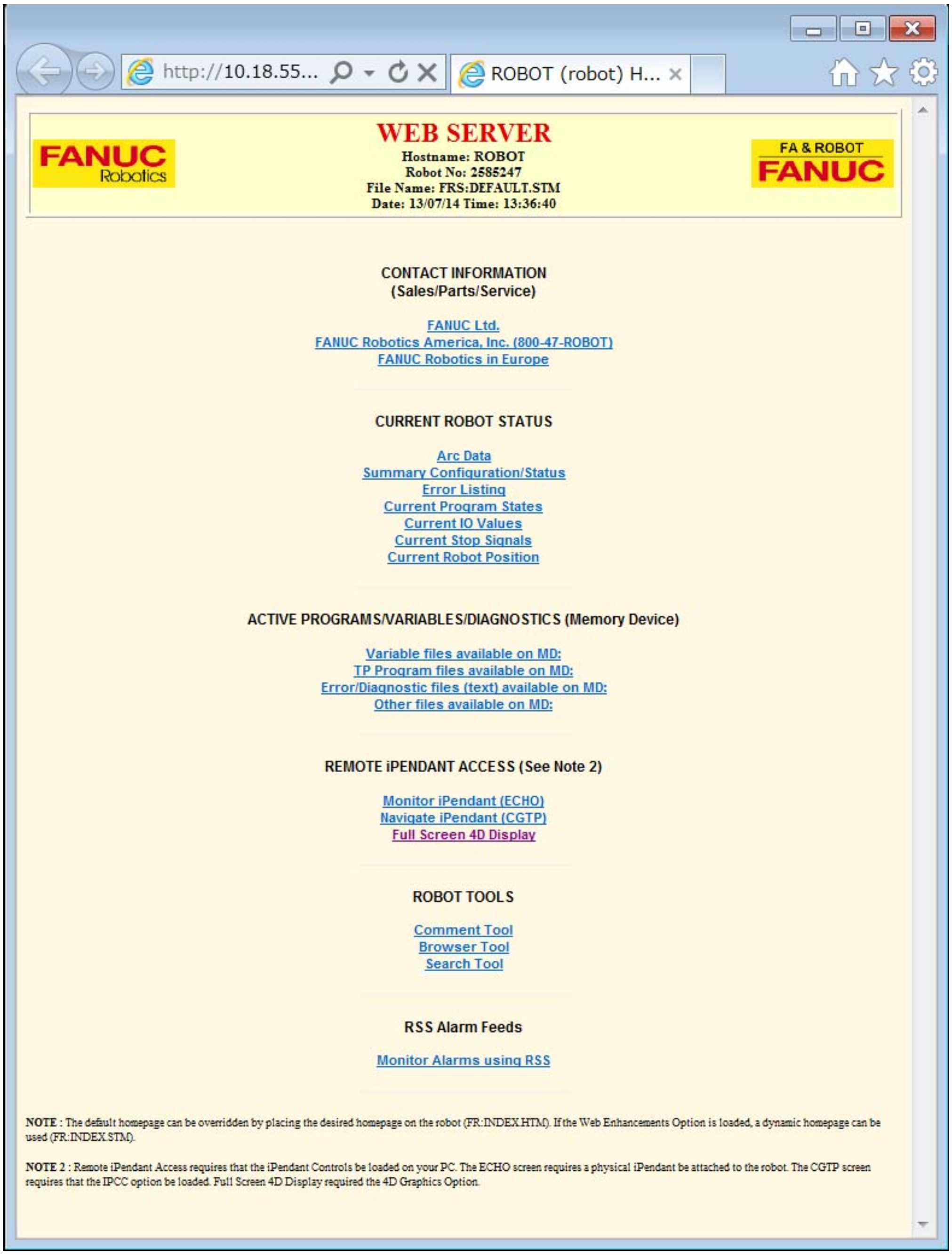
This section will describe the method to connect to the robot controller and display the Full Screen 4D Display screen.

Procedure 37-8 Accessing The Full Screen 4D Display

Steps

1 Bring up Internet Explorer on the PC.

2 In the Internet Explorer Address field, type the following: http://<myrobot_name_ or_address> to view the robot HOME page.
Where <myrobot_name_ or_address> is either the DNS name of your robot (i.e. pderob111.frc.com) or the IP address of your robot (i.e. 192.168.1.100).
If the item is "Full Screen 4D Display : Unavailable", check if 4D Graphics (R764) option is ordered.
If the item is not displayed, check if Internet Conn/Custo (R558) option is ordered.



WEB SERVER
Hostname: ROBOT
Robot No: 2585247
File Name: FRS:DEFAULT.STM
Date: 13/07/14 Time: 13:36:40

CONTACT INFORMATION
(Sales/Parts/Service)

FANUC Ltd.
FANUC Robotics America, Inc. (800-47-ROBOT)
FANUC Robotics in Europe

CURRENT ROBOT STATUS

Arc Data
Summary Configuration/Status
Error Listing
Current Program States
Current IO Values
Current Stop Signals
Current Robot Position

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)
Full Screen 4D Display

ROBOT TOOLS

Comment Tool
Browser Tool
Search Tool

RSS Alarm Feeds

Monitor Alarms using RSS

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).

NOTE 2 : Remote iPendant Access requires that the iPendant Controls be loaded on your PC. The ECHO screen requires a physical iPendant be attached to the robot. The CGTP screen requires that the IPCC option be loaded. Full Screen 4D Display required the 4D Graphics Option.

Fig. 37.4.3.1(a) Robot Home Page

3 From the HOME page, when you select Full Screen 4D Display, 4D graphics display screen similar to the one shown in Figure 37.4.3.1(b) is displayed on your PC.

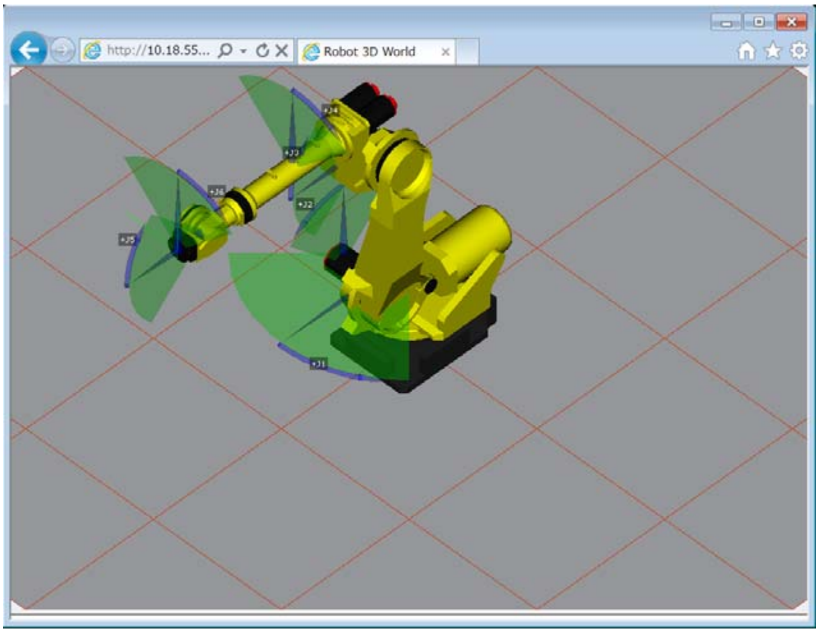


Fig. 37.4.3.1(b) Full Screen 4D Display Screen

37.4.3.2 View Adjustment Mode

The view of the 4D scene can be adjusted with the mouse. The mouse can PAN, ROTATE or ZOOM the view.

Procedure 37-9 Selecting View Adjustment Mode

Steps

1 Click with the right (alternate) mouse button.

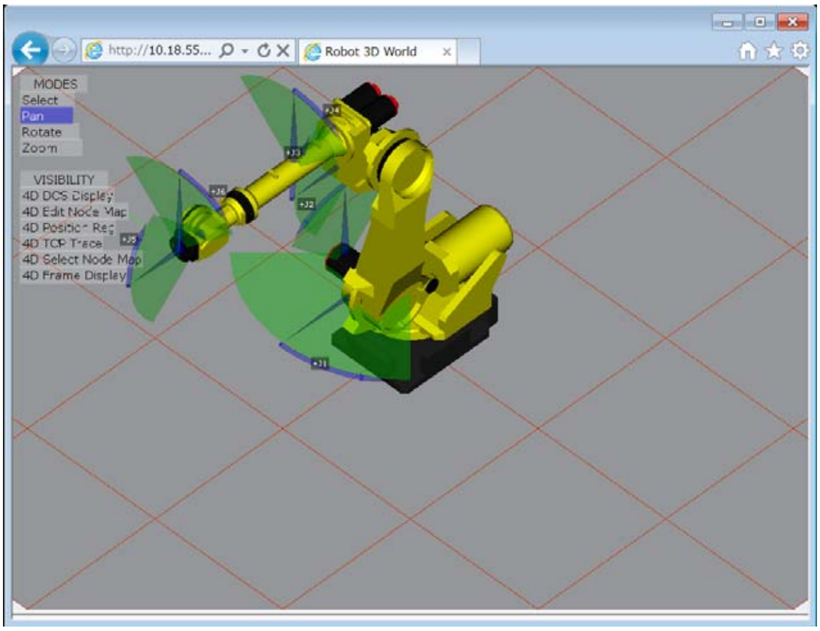


Fig. 37.4.3.2 Full Screen 4D Display Screen

2 Use either mouse button to select the view adjustment mode you want:

- Pan - To move the view up and down and side to side.
- Rotate - To rotate the view.
- Zoom - To zoom the view in and out.

Click anywhere outside the menu to dismiss the menu.

3 Use the left (primary) mouse button to adjust the view according to the selection.

37.4.3.3 Scene Visibility

Scene visibility allows selection of any combination of 4D display entities. The highlighted (white on blue) elements in the menu are the ones which are currently being displayed. Selecting an element will toggle the visibility on or off.

Procedure 37-10 Selecting Scene Visibility

Steps

1 Click with the right (alternate) mouse button.

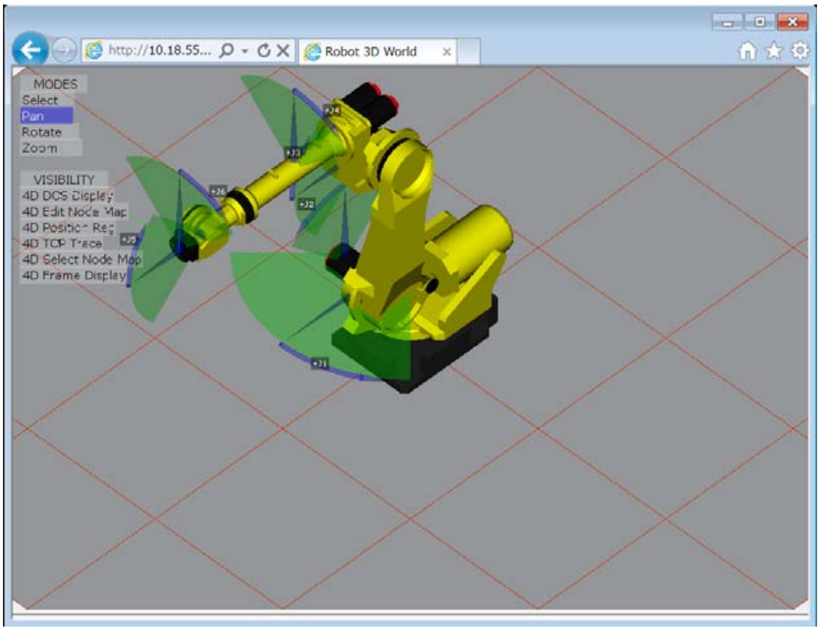


Fig. 37.4.3.3(a) 4D Graphics Node Map Scene

2 Use the left or right mouse button to select the scene you want to toggle:

- To change visible/invisible status of each data, select them.
- When you select multiple items in the scene information, the all selected scene will be displayed in 4D screen.

Please refer to section 37.2 4D GRAPHICS SCENE for details of each scene.

For example, at the system like Fig. 37.4.3.3, when you select 4D Select Node Map, the node map is displayed on your PC.

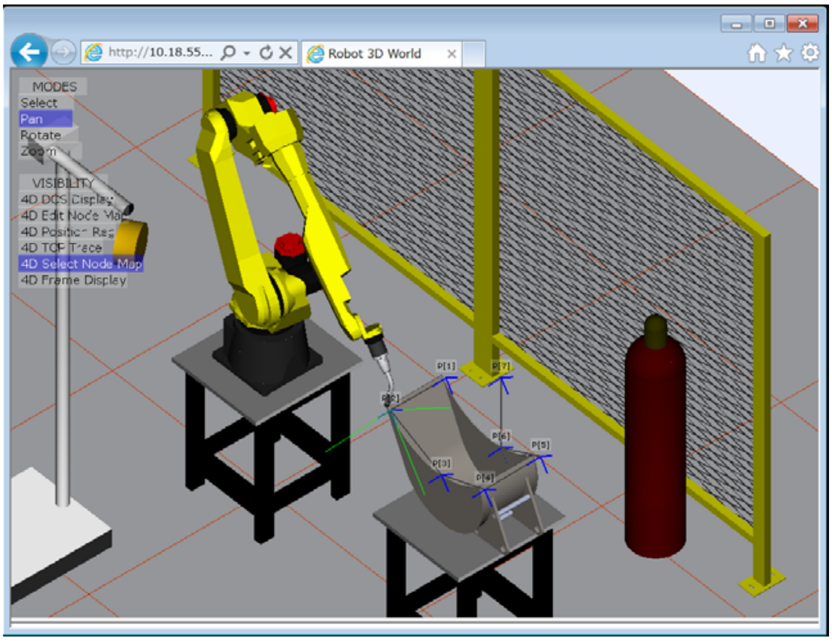


Fig. 37.4.3.3(b) 4D Graphics Node Map Scene

3 Click anywhere outside the menu to dismiss the menu

NOTE

If DCS Pos./Speed check (J567) option is not loaded, the item “4D DCS Display” is not displayed.

38 DATA TRANSFER BETWEEN ROBOTS FUNCTION

This section explains data transfer between robots function. This function is optional (J740).

This function enables you to transfer data between robots over Ethernet.
By calling KAREL program, you can transfer register and position register between robot controllers.

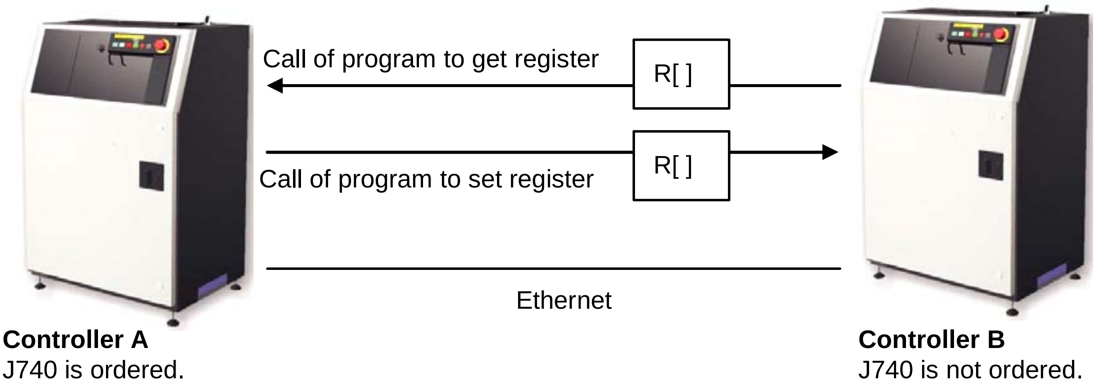


Fig. 38 Data transfer between robots

In Fig. 38, controller A runs program to get register from controller B. In addition, Controller A call program to transfer register to controller B.
In Fig. 38, only controller A starts data transfer. Controller B is just responding to request from controller A. In this case, only controller A needs this option.

> **NOTE**
> In case that robot controller to communicate with (controller B in Fig. 38.1) is R-30*i*A or R-30*i*A Mate controller, series of the system software of that controller must be 7DA5 or later.

This function also provides KAREL built-in for data transfer. By KAREL programming you can transfer register and comment of I/O of remote controller. You can also read and write value of I/O. KAREL is program language for robot. It is used to for construction of robot system. For more detail on KAREL, please refer to “R-30*i*A/R-30*i*A Mate controller KAREL function OPERATOR’S MANUAL” (B-83144EN). Please refer to “38.7 KAREL built-in” for built-in routines provided by this function.

38.1 TERMINOLOGY

To simplify and clarify explanation of this function we use following term.

Client

Client is a robot controller that starts communication by this function. In Fig. 38.1, controller A is client. Client runs program to request various service to another controller.

Server

Server is a robot controller that receives request from client and serve for it. In Fig. 38.1, controller B is server.

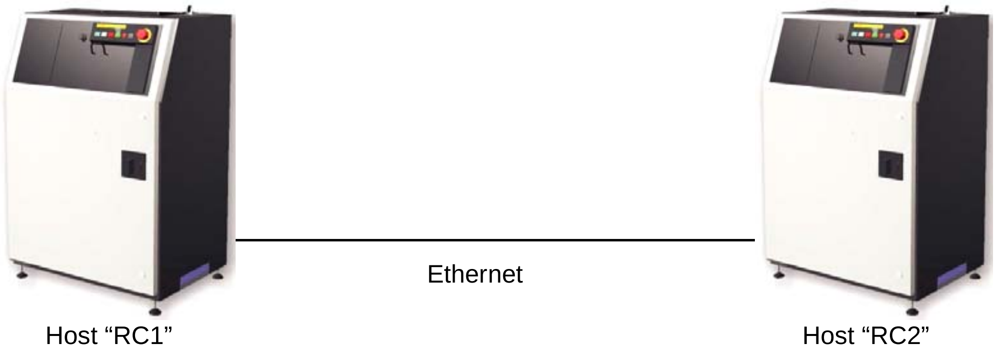
38.2 SETUP

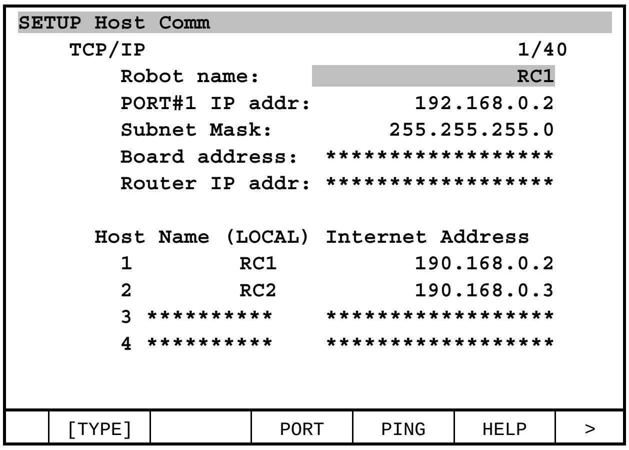
Connect robot controllers by Ethernet. Set TCP/IP parameters so that controllers can communicate over Ethernet by TCP/IP.

1 Press [MENU] key.
2 Select SETUP.
3 Select Host Comm.
4 Input host name (Robot name) and IP address and those of controllers to communicate with.
5 Cycle power.

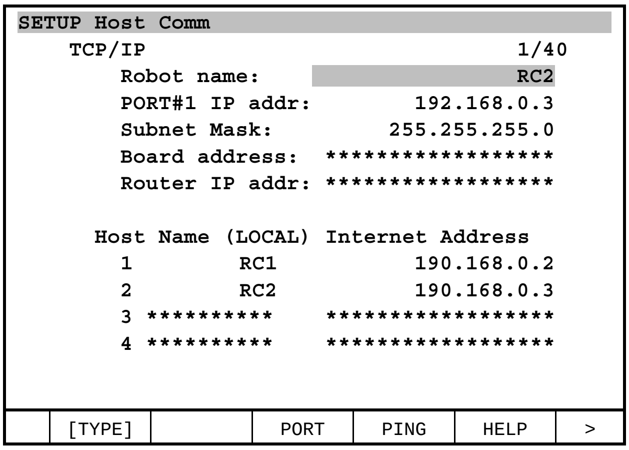
For more detail, please refer to “R-30*i*A/R-30*i*A Mate/R-30*i*B/R-30*i*B Mate Ethernet Function OPERATOR’S MANUAL (B-82974EN)”, Chapter2 “Setting up TCP/IP”.

> **NOTE**
> Don’t use underscore (“_”) in host name.





```
SETUP Host Comm
   TCP/IP                              1/40
       Robot name:                      RC1
       PORT#1 IP addr:          192.168.0.2
       Subnet Mask:           255.255.255.0
       Board address:  ******************
       Router IP addr: ******************

     Host Name (LOCAL) Internet Address
       1         RC1            190.168.0.2
       2         RC2            190.168.0.3
       3 **********    ******************
       4 **********    ******************
```

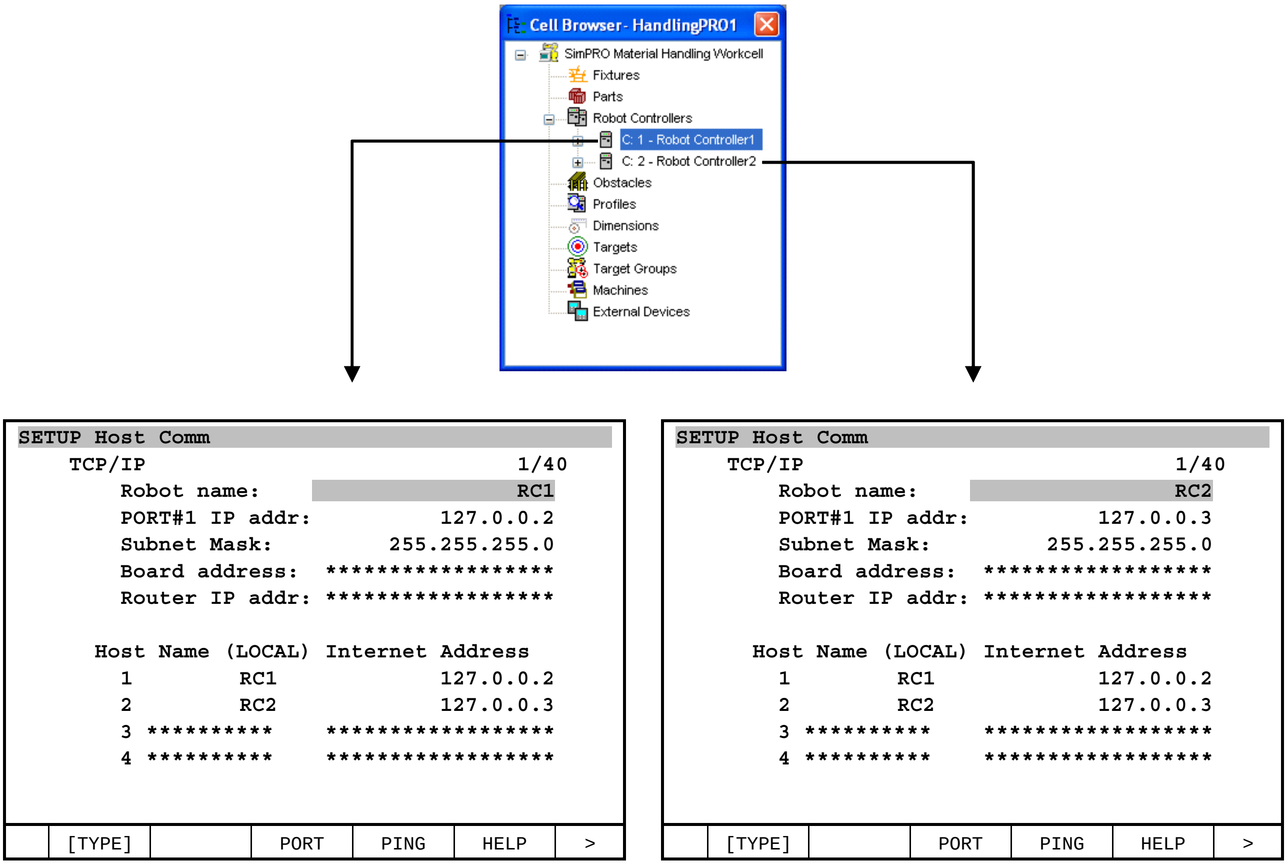


```
SETUP Host Comm
   TCP/IP                              1/40
       Robot name:                      RC2
       PORT#1 IP addr:          192.168.0.3
       Subnet Mask:           255.255.255.0
       Board address:  ******************
       Router IP addr: ******************

     Host Name (LOCAL) Internet Address
       1         RC1            190.168.0.2
       2         RC2            190.168.0.3
       3 **********    ******************
       4 **********    ******************
```

38.3 TCP/IP SETUP FOR ROBOGUIDE

You can simulate data transfer in a work cell of ROBOGUIDE. Please note that IP address of each host must be 127.0.0.x.

Following figure is an example of TCIP/IP setup for ROBOGUIDE. There are two robot controllers in your work cell. Host name of the 1st one is RC1 and the 2nd one is RC2.



```
SETUP Host Comm
    TCP/IP                                1/40
        Robot name:                        RC1
        PORT#1 IP addr:              127.0.0.2
        Subnet Mask:             255.255.255.0
        Board address:      ******************
        Router IP addr:     ******************

      Host Name (LOCAL) Internet Address
        1         RC1                127.0.0.2
        2         RC2                127.0.0.3
        3 **********        ******************
        4 **********        ******************

    [TYPE]          PORT    PING    HELP    >
```

```
SETUP Host Comm
    TCP/IP                                1/40
        Robot name:                        RC2
        PORT#1 IP addr:              127.0.0.3
        Subnet Mask:             255.255.255.0
        Board address:      ******************
        Router IP addr:     ******************

      Host Name (LOCAL) Internet Address
        1         RC1                127.0.0.2
        2         RC2                127.0.0.3
        3 **********        ******************
        4 **********        ******************

    [TYPE]          PORT    PING    HELP    >
```

You have to re-start robot controllers after TCP/IP setup.

If you run following program, value and comment of R [1] of RC1 is transferred to R [5] of RC2.

```
   1:  CALL RGETNREG('RC1',1,5,0) ;
```

RGETNREG is KAREL program this function provides. For more detail, please refer to Section 38.4 STANDARD DATA TRANSFER PROGRAM.

38.4 STANDARD DATA TRANSFER PROGRAM

This function includes programs to transfer register and position register. Following table lists the programs.

Program	Function	Syntax
RGETNREG	To get numeric register	RGETNREG('host/IP address', source index, destination index, option)
RSETNREG	To set numeric register	RSETNREG('host/IP address', destination index, source index, option)
RGETPREG	To get position register	RGETPREG('host/IP address', source index, source group number, destination index, destination group, option)
RSETPREG	To set position register	RSETPREG('host/IP address', destination index, destination group, source index, source group number, option)

We describe each program in subsection 38.5.1 and later.

NOTE

If arguments are not proper, it causes error. Followings are example of wrong arguments.

- Type of argument (integer, string and the like) is wrong.
- The number of arguments is less than expected.
- Content of argument is wrong.

Please call program with proper argument.

When you restart execution of program, please move cursor to 1 line before CALL and restart from the line.

38.4.1 Program to Get Numeric Register

Program name: RGETNREG

Overview

Client controller gets register from server and sets it to register of client.

Syntax

RGETNREG (host_name, src_idx, dest_idx, option)
Input/Output parameters
The 1st parameter: [IN] String of host name or IP address of server.
The 2nd parameter: [IN] Index of source register. This integer is index of register of server.
The 3rd parameter: [IN] Index of destination register. This integer is index of register of client.
The 4th parameter: [IN] This integer specifies function of this program.

Value	Description
0	Value and comment are got.
1	Value is got.
2	Comment is got.

Detail

If there isn't R [src_idx] on server, "DTBR-014 Bad variable or register index (host name, index)" is posted.
If there isn't R [dest_idx] on client, "VARS-024 Bad variable or register index" is posted.

Example

Execution of following program gets value and comment of R [10] of Host "SERVER".

```
CALL RGETNREG ('SERVER', 10, 20, 0)
```

38.4.2 Program to Set Numeric Register

Program name: RSETNREG

Overview

Client controller sets client's register to server's register.

Syntax

RSETNREG (host_name, dest_idx, src_idx, option)
Input/Output parameters
The 1st parameter: [IN] String of host name or IP address of server.
The 2nd parameter: [IN] Index of destination register. This integer is index of register of server.
The 3rd parameter: [IN] Index of source register. This integer is index of register of client.
The 4th parameter: [IN] This integer specifies function of this program.

Value	Destination
0	Value and comment are set.
1	Value is set.

Detail

If there isn’t R [dest_idx] on server, “DTBR-014 Bad variable or register index (host name, index)” is posted.
If there isn’t R [src_idx] on client, “VARS-024 Bad variable or register index" is posted.

Example

Execution of following program sets value and comment of R [20] of client to R [10] of host “SERVER”.

```
CALL RSETNREG (‘SERVER’, 10, 20, 0)
```

38.4.3 Program to Get Position Register

Program name: RGETPREG

Overview

Client controller gets position register from server and set it to position register of client. Position data of specified group only is got.

Syntax

RGETPREG (host_name, src_idx, src_grp, dest_idx, dest_grp, option)
Input/Output parameters
The 1st parameter: [IN] String of host name or IP address of server.
The 2nd parameter: [IN] Index of source position register. This integer is index of PR of server.
The 3rd parameter: [IN] Group number of source position register. This integer is group number of server.
The 4th parameter: [IN] Index of destination position register. This integer is index of PR of client.
The 5th parameter: [IN] Group number of destination position register. This integer is group number of client.
The 6th parameter: [IN] This integer specifies function of this program.

Value	Description
0	Value and comment are got.
1	Value is got.
2	Comment is got.

Detail

If contents of the 2nd through the 5th parameters are wrong, following errors are posted.

Error	Cause
DTBR-014 Bad variable or register index (host name, index)	There isn’t PR [(the 2nd parameter) src_idx] on server.
DTBR-015 Illegal group number (host name, index)	There isn’t group “(the 3rd parameter) src_grp” on server.
VARS-024 Bad variable or register index	There isn’t PR [(the 4th parameter) dest_idx] on client.
ROUT-026 Illegal group number	There isn’t group “(the 5th parameter)dest_grp”on client.

NOTE

Position data of specified groups of client and server may be different in its components. For example, following points may differ.

- The number of axes
- Configuration of Cartesian position
- Type of axis (rotary or linear)

You may not be able to use position register that is got or set as it is. Please be careful to use transferred data.

NOTE

Even if Cartesian position data is got and stored to client's group that does not support Cartesian data (independent axis for example), Cartesian data is set.

Example

Execution of following program by client transfers position data of group 2 of PR [10] of host "SERVER" to client's group 1 of PR[20]. Comment is not acquired.

```
CALL RGETPREG ('SERVER', 10, 2, 20, 1)
```

38.4.4 Program to Set Position Register

Program name: RSETPREG

Overview

Client controller sets client's position register to server's position register. Position data of specified group only is set.

Syntax

RSETPREG (host_name, dest_idx, dest_grp, src_idx, src_grp, option)
Input/Output parameters
The 1st parameter: [IN] String of host name or IP address of server.
The 2nd parameter: [IN] Index of destination position register. This integer is index of PR of server.
The 3rd parameter: [IN] Group number of position data of destination PR of server. This is INTEGER.
The 4th parameter: [IN] Index of source position register. This integer is index of PR of client.
The 5th parameter: [IN] Group number of position data of source PR of client. This is INTEGER.
The 6th parameter: [IN] This integer specifies function of this program.

Value	Description
0	Position data and comment is set.
1	Position data is set.

Detail

If contents of the 2nd through 5th parameters are wrong, following errors are posted.

Error	Cause
DTBR-014 Bad variable or register index (host name, index)	There isn't PR [(the 2nd parameter) dest_idx] on server.
DTBR-015 Illegal group number(host name, index)	There isn't group"(the 3rd parameter) dest_grp" on server.
VARS-024 Bad variable or register index	There isn't PR [(the 4th parameter) src_idx] on client.
ROUT-026 Illegal group number	There isn't group "(the 5th parameter) src_grp" on client.

NOTE

Position data of specified groups of client and server may be different in its components. For example, following points may differ.

- The number of axes
- Configuration of Cartesian position
- Type of axis (rotary or linear)

You may not be able to use position register that is got or set as it is. Please be careful to use transferred data.

NOTE

Even if Cartesian position data is set to server's group that does not support Cartesian data (independent axis for example), Cartesian data is set.

Example

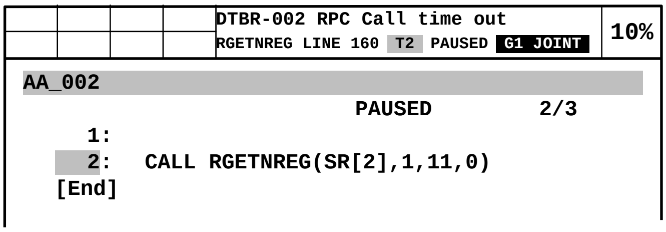
Execution of following program by client transfers position data of group 1 of PR [20] of client to group 2 of PR [10] of host "SERVER".

```
CALL RSETPREG ('SERVER', 10, 2, 20, 1, 0)
```

38.5 RECOVERY FROM ERROR

When error happened at execution of standard data transfer program, most of errors are following 2 cases.

1 "DTBR-002 RPC Call timed out" is posted because communication failed or was not in time.



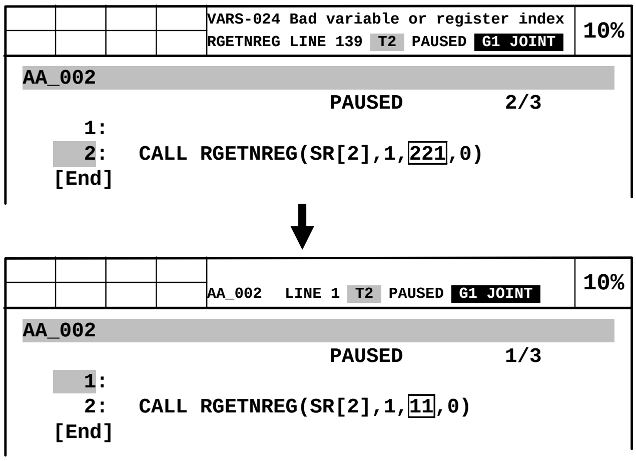
```
DTBR-002 RPC Call time out
RGETNREG LINE 160 T2 PAUSED G1 JOINT          10%
AA_002
                              PAUSED        2/3
    1:
    2:  CALL RGETNREG(SR[2],1,11,0)
  [End]
```

Please check if the controller (client) can always, stably communicate with specified host (server). You can perform "PING" in PING screen of Host Comm setup screen.
Please improve network traffics.
If standard data transfer program or built-in of this function are used very often, please lessen frequency.
If standard data transfer program or built-in of this function is executed consecutively, there should be interval. If there is already interval, please lengthen it.

When program is resumed without moving cursor, the same transfer is performed again.
If parameter and TCIP/IP setup has no problem and network had trouble, resume after removal of network problem can be successful.

2 Arguments were wrong.
In this case, arguments must be modified. In program edit screen, cursor is on the call of standard data transfer program. However, execution stopped inside of called program. Arguments were already handed to sub program (in figure below, RGETNREG).
Even if you modify arguments in program edit screen, resume from the same line doesn't solve the problem. It is because value of arguments was already handed to sub program. If arguments are register or string register, value was handed at call. Even if you change value of register or string register after error happened, the change is not reflected to resume from the same line.

After modification of arguments, please move cursor to 1 line before CALL and resume from the line.



```
VARS-024 Bad variable or register index
RGETNREG LINE 139  T2  PAUSED  G1 JOINT        10%
AA_002
                         PAUSED         2/3
     1:
     2:   CALL RGETNREG(SR[2],1,221,0)
   [End]
```

↓

```
AA_002   LINE 1  T2  PAUSED  G1 JOINT          10%
AA_002
                         PAUSED         1/3
     1:
     2:   CALL RGETNREG(SR[2],1,11,0)
   [End]
```

38.6 KAREL BUILT-IN

KAREL built-in routine (or function) is pre-defined function, which is built in KAREL language. This function provides built-in routines to transfer data between robots. KAREL program using the built-in enables followings.

- Sending and receiving register
- Sending and receiving position register
- Sending and receiving string register
- Read and write of I/O value
- Read of I/O simulation status
- Simulating and un-simulating I/O
- Read and write of comment of following data
    - Register
    - Position register
    - String register
    - I/O

Following table lists built-in routines provided by this function.

Built-in	Description
RGET_PORTCMT	gets I/O comment.
RGET_PORTSIM	gets I/O simulation status.
RGET_PORTVAL	gets I/O value.
RGET_PREGCMT	gets comment of position register.
RGET_REG	gets register.
RGET_REG_CMT	gets comment of register.
RGET_SREGCMT	gets comment of string register.
RGET_STR_REG	gets string register.
RNREG_RECV	transfers server’s register to client’s register.
RNREG_SEND	transfers client’s register to server’s register.
RPREG_RECV	transfers server’s position register to client’s position register.
RPREG_SEND	transfers client’s position register to server’s position register.
RSET_INT_REG	sets INTEGER to register.
RSET_PORTCMT	sets I/O comment.
RSET_PORTSIM	simulates I/O.
RSET_PORTVAL	sets I/O value.
RSET_PREGCMT	sets comment of position register.

Built-in	Description
RSET_REALREG	sets REAL value to register.
RSET_REG_CMT	sets comment of register.
RSET_SREGCMT	sets comment of string register.
RSET_STR_REG	sets string register.

38.6.1 RGET_PORTCMT Built-in ROUTINE

Purpose:

To allows a KAREL program to determine comment that is set for a specified logical port of remote host.

Syntax:

RGET_PORTCMT (host_port, port_type, port_no, comment_str, status)

Input/Output parameters

[in] host_port: STRING
[in] port_type: INTEGER
[in] port_no: INTEGER
[out] comment_str: STRING
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “port_type” specifies the code for the type of port whose comment is returned. Codes are defined in kliotyps.kl.
- “port_no” specifies the port number whose comment is being returned.
- “comment_str” is returned with the comment for the specified port. This should be declared as a STRING with a length of at least 16 characters. Only 16 characters are returned even if server’s comment is longer.
- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program gets comment of DO[10] of host RC1.

```
PROGRAM RGTPCMT
%COMMENT='RC1 DO[10] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%INCLUDE kliotyps

VAR
  status  : INTEGER
  comment : STRING[16]
BEGIN
  RGET_PORTCMT('RC1', io_dout, 10, comment, status)
END RGTPCMT
```

38.6.2 RGET_PORTSIM Built-in ROUTINE

Purpose:

To get port simulation status from remote controller.

Syntax:

RGET_PORTSIM (host_port, port_type, port_no, simulated, status)
Input/Output parameters
[in]host_port: STRING
[in]port_type: INTEGER

[in]port_no: INTEGER
[out]simulated: BOOLEAN
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “port_type” specifies the code for the type of port whose simulation status is returned. Codes are defined in kliotyps.kl.
- “port_no” specifies the port number whose simulation status is being returned.
- “simulated” returns TRUE if the port is being simulated, FALSE otherwise.
- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program gets simulation status of DI[8] of host RC1.

```
PROGRAM RGTPSIM
%COMMENT = 'RC1 DIN[8] SIM'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%include kliotyps
VAR
  status  : INTEGER
  sim_stat: BOOLEAN
BEGIN
  RGET_PORTSIM('RC1', io_din, 8, sim_stat, status)
END RGTPSIM
```

38.6.3 RGET_PORTVAL Built-in ROUTINE

Purpose:

To allow KAREL program to determine the current value of a specified logical port of remote host.

Syntax:

RGET_PORTVAL (host_port, port_type, port_no, port_value, status)
Input/Output parameters
[in]host_port: STRING
[in]port_type: INTEGER
[in]port_no:INTEGER
[out]port_value:INTEGER
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “port_type” specifies the code of port whose value is being returned. Codes are defined in kliotyps.kl.
- “port_no” specifies the port number whose value is returned.
- “port_value” is returned with the current value (status) of the specified port. For BOOLEAN port types (DIN for example), this will be 0 = OFF, or 1 = ON.
- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program gets value of RO [3] of host RC1.

```
PROGRAM RGTPVAL
%COMMENT='GET RC1 RO[3]'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%include kliotyps
VAR
  status  : INTEGER
  port_val: INTEGER
BEGIN
  RGET_PORTVAL('RC1', io_rdo, 3, port_val, status)
END RGTPVAL
```

38.6.4 RGET_PREGCMT Built-in ROUTINE

Purpose:

To retrieve comment of a position register of remote host.

Syntax:

RGET_PREGCMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[out]comment_str : STRING
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “register_no” specifies which position register to retrieve the comment from. The comment of the given position register is returned in parameter “comment_str”.
- “comment_str” should be declared as a STRING with a length of at least 16 characters.
- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program gets comment of PR [5] of host RC1.

```
PROGRAM RGTPRCMT
%COMMENT='RC1 PR[5] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
  comment : STRING[16]
BEGIN
  RGET_PREGCMT('RC1', 5, comment, status)
END RGTPRCMT
```

38.6.5 RGET_REG Built-in ROUTINE

Purpose:

To get an INTEGER or REAL value from the specified register of remote host.

Syntax:

RGET_REG (host_port, register_no, real_flag, int_value, real_value, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[out]real_flag: BOOLEAN

[out]int_value: INTEGER
[out]real_value: REAL
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “register_no” specifies the register to get.
- “real_flag” is set to TRUE and “real_value” is set to the register content if the specified register has a real value. Otherwise, “real_flag” is set to FALSE and “int_value” is set to the contents of the register.
- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program gets value of R [2] of host RC1.

```
PROGRAM RGTNREG
%COMMENT='GET RC1 R[2]'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status   : INTEGER
  is_real  : BOOLEAN
  int_val  : INTEGER
  real_val : REAL
BEGIN
  RGET_REG('RC1', 2, is_real, int_val, real_val, status)
END RGTNREG
```

38.6.6 RGET_REG_CMT Built-in ROUTINE

Purpose:

To get comment from the specified register of remote host.

Syntax:

RGET_REG_CMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in] host_port: STRING
[in] register_no: INTEGER
[out] comment_str: STRING
[out] status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “register_no” specifies which register to retrieve the comment from. The comment of the given register is returned in “comment_str”.
- “comment_str” should be declared as a STRING with a length of at least 16 characters.
- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program gets comment of R [2] of host RC1.

```
PROGRAM RGTNRCMT
%COMMENT='GET RC1 R[2] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
  comment : STRING[16]
BEGIN
  RGET_REG_CMT('RC1', 2, comment, status)
END RGTNRCMT
```

38.6.7 RGET_SREGCMT Built-in ROUTINE

Purpose:

To get comment from the specified string register of remote host.

Syntax:

RGET_SREGCMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[out]comment_str: STRING
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “register_no” specifies which string register to retrieve the comment from.
- “comment_str” contains the comment of the specified string register.
- “comment_str” should be declared as a STRING with a length of at least 16 characters.
- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Examples:

Following program gets comment of SR [20] of host RC1.

```
PROGRAM RGTSRCMT
%NOLOCKGROUP
%COMMENT = 'GET CMT SR[20]'
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
  comment : STRING[16]
BEGIN
  RGET_SREGCMT('RC1', 20, comment, status)
END RGTSRCMT
```

38.6.8 RGET_STR_REG Built-in ROUTINE

Purpose:

To get value from the specified string register of remote host.

Syntax:

RGET_STR_REG (host_port, register_no, value, status)
Input/Output parameters
[in] host_port: STRING
[in] register_no: INTEGER
[out] value : STRING
[out] status: INTEGER

%ENVIRONMENT Group: RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “register_no” specifies string register to get.
- “value” contains the value of the specified string register.
- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program gets value of SR [10] or host RC1.

```
PROGRAM RGTSREG
%NOLOCKGROUP
%ENVIRONMENT RPCC
%COMMENT = 'GET SR[10]'
VAR
  status   : INTEGER
  sreg_val : STRING[253]
BEGIN
  RGET_STR_REG('RC1', 10, sreg_val, status)
END RGTSREG
```

38.6.9 RNUMREG_RECV Built-in ROUTINE

Purpose:

To transfer server’s register to client’s register.

Syntax:

RNUMREG_RECV (host_port, src_idx, dest_idx, option, status)
Input/Output parameters
[in]host_port: STRING
[in]src_idx : INTEGER
[in]dest_idx : INTEGER
[in]option : INTEGER
[in]status : INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “src_idx” specifies index of register of server.
- “dest_idx” specifies index of register of client. Acquired data is stored in this register.
- “option” specifies function of this built-in.

Value	Description
0	Value and comment are got.
1	Value is got.
2	Comment is got.

- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program gets value and comment of R [3] of RC1 and set them to R [12].

```
PROGRAM RNREGRCV
%COMMENT='RC1 R3->12'
%NOLOCKGROUP
%ENVIRONMENT RPCC

VAR
  STATUS : INTEGER
BEGIN
  RNUMREG_RECV('RC1', 3, 12, 0, STATUS)
END RNREGRCV
```

38.6.10 RNUMREG_SEND Built-in ROUTINE

Purpose:

To transfer client’s register to server’s register.

Syntax:

RNUMREG_SEND (host_port, dest_idx, src_idx, option, status)
Input/Output parameters
[in]host_port: STRING
[in]dest_idx : INTEGER
[in]src_idx : INTEGER
[in]option : INTEGER
[in]status : INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “dest_idx” specifies index of register of server.
- “src_idx” specifies index of register of client. Data of this register is sent to server.
- “option” specifies function of this built-in.

Value	Descrition
0	Value and comment is set.
1	Value is set.
2	Comment is set.

- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sends value of R [11] to R [2] of host RC1.

```
PROGRAM RNREGSND
%COMMENT='R11->RC1 R1'
%NOLOCKGROUP
%ENVIRONMENT RPCC

VAR
  STATUS : INTEGER
BEGIN
  RNUMREG_SEND('RC1', 2, 11, 1, STATUS)
END RNREGSND
```

38.6.11 RPOSREG_RECV Built-in ROUTINE

Purpose:

To transfer position register of specified group of server to position register of specified group of client.

Syntax:

RPOSREG_RECV (host_port, src_idx, src_grp, dest_idx, dest_grp, option, status)
Input/Output parameters
[in]host_port: STRING
[in]src_idx: INTEGER
[in]src_grp: INTEGER
[in]dest_idx: INTEGER
[in]dest_grp: INTEGER
[in]option: INTEGER
[in]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "src_idx" specifies index of position register of server.
- "src_grp" specifies group number of server. Position data of specified group is transferred.
- "dest_idx" specifies index of position register of client. The specified position register stores acquired data.
- "dest_grp" specifies group number of client. Position data of specified group of PR ["dest_idx"] is changed to received data.
- "option" specifies function of this built-in.

Value	Description
0	Position data and comment are got.
1	Position data is got.
2	Comment is got.

- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following example program performs following transfer. Comment is transferred, too.
From: Group 1 of PR [5] of host RC1
To: Group 2 of PR [6]

```
PROGRAM RPREGRCV
%COMMENT='RC1 PR5G1->PR6G2'
%NOLOCKGROUP
%ENVIRONMENT RPCC

VAR
  STATUS : INTEGER
BEGIN
  RPOSREG_RECV('RC1', 5, 1, 6, 2, 0, STATUS)
END RPREGRCV
```

38.6.12 RPOSREG_SEND Built-in ROUTINE

Purpose:

To transfer position register of specified group of client to position register of specified group of server.

Syntax:

RPOSREG_SEND (host_port, dest_idx, dest_grp, src_idx, src_grp, option, status)
Input/Output parameters
[in]host_port: STRING
[in]dest_idx: INTEGER
[in]dest_grp: INTEGER
[in]src_idx: INTEGER

[in]src_grp: INTEGER
[in]option: INTEGER
[in]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “dest_idx” specifies index of position register of server.
- “dest_grp” specifies group number of server. Position data of specified group is changed by sent data from client.
- “src_idx” specifies index of position register of client. Position data of “src_grp” of PR [“src_idx”] is sent to server.
- “src_grp” specifies group number of client.
- “option” specifies function of this built-in.

Value	Description
0	Position data and comment are set.
1	Position data is set.

- status explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following example program performs following transfer. Comment is transferred, too.
From: Group 2 of PR [7] of client
To: Group1 of PR [8] of host RC1

```
PROGRAM RPREGSND
%COMMENT='PR7G2->RC1 PR8G1'
%NOLOCKGROUP
%ENVIRONMENT RPCC

VAR
  STATUS : INTEGER
BEGIN
  RPOSREG_SEND('RC1', 8, 1, 7, 2, 0, STATUS)
END RPREGSND
```

38.6.13 RSET_INT_REG Built-in ROUTINE

Purpose:

To store an INTEGER value in the specified register of remote host.

Syntax:

RSET_INT_REG (host_port, register_no, int_value, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[in]int_value: INTEGER
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “register_no” specifies the register into which “int_value” will be stored.
- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sets 100 to R[20] of host RC1.

```
PROGRAM RSTNRI
%COMMENT = 'RC R[20]=100'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status : INTEGER
BEGIN
  RSET_INT_REG('RC1', 20, 100, status)
END RSTNRI
```

38.6.14 RSET_PORTCMT Built-in ROUTINE

Purpose:

To allows a KAREL program to set comment of specified logical port of remote host.

Syntax:

RSET_PORTCMT (host_port, port_type, port_no, comment_str, status)
Input/Output parameters
[in]host_port: STRING
[in]port_type: INTEGER
[in]port_no: INTEGER
[in]comment_str: STRING
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “port_type” specifies the code of port whose comment is being set. Codes are defined in kliotyps.kl.
- “port_no “specifies the port number whose comment is being set.
- “comment_str” is a string whose value is the comment for the specified port. This must not be over 16 characters long.
- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sets comment to DO [10] of host RC1.

```
PROGRAM RSTPCMT
%COMMENT ='RC1 DO[10] cmt'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%INCLUDE kliotyps
VAR
  status : INTEGER
BEGIN
  RSET_PORTCMT('RC1', io_dout, 10, 'RC1 DOUT[10]', status)
END RSTPCMT
```

38.6.15 RSET_PORTSIM Built-in ROUTINE

Purpose:

To set port simulated on remote host.

Syntax:

RSET_PORTSIM (host_port, port_type, port_no, value, status)
Input/Output parameters

[in]host_port: STRING
[in]port_type: INTEGER
[in]port_no: INTEGER
[in]value: INTEGER
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "port_type" specifies the code for the type of port to be simulated. Codes are defined in kliotyps.kl.
- "port_no" specifies port number to be simulated.
- "value" specifies the initial value to set.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sets GO [3] of RC1 simulated and set its initial simulated value to 5.

```
PROGRAM RSTPSIM
%COMMENT ='RC1 SIM GO[3], 5'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%INCLUDE kliotyps

VAR
  status  : INTEGER
BEGIN
  RSET_PORTSIM('RC1', io_gpout, 3, 5, status)
END RSTPSIM
```

38.6.16 RSET_PORTVAL Built-in ROUTINE

Purpose:

To allow KAREL program to set a specified output (or simulated input) for a specified logical port.

Syntax:

RSET_PORTVAL (host_port, port_type, port_no, port_value, status)
Input/Output parameters
[in]host_port: STRING
[in]port_type: INTEGER
[in]port_no: INTEGER
[in]port_value: INTEGER
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "port_type" specifies the code of port whose value is being set. Codes are defined in kliotyps.kl.
- "port_no" specifies the port number whose value is being set.
- "value" indicates the value to be assigned to a specified port. If the port_type is BOOLEAN (DOUT for example), this should be 0 = OFF, or 1 = ON. This field can be used to set input ports if the port is simulated.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sets DO [3] of host RC1 to ON.

```
PROGRAM RSTPVAL
%COMMENT ='RC1 DO[3]=ON'
%NOLOCKGROUP
%ENVIRONMENT RPCC
%INCLUDE kliotyps

VAR
  status  : INTEGER
BEGIN
  RSET_PORTVAL('RC1', io_dout, 3, 1, status)
END RSTPVAL
```

38.6.17 RSET_PREGCMT Built-in ROUTINE

Purpose:

To set comment of a position register of remote host.

Syntax:

RSET_PREGCMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[in]comment_str : STRING
[out]status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “register_no” specifies position register of remote robot controller.
- “comment_str” is comment to be set. If “comment_str” exceeds 16 characters, it is truncated.
- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sets comment “RC1 PR[5]” to PR [5] of host RC1.

```
PROGRAM RSTPRCMT
%COMMENT ='RC1 PR[5] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
BEGIN
  RSET_PREGCMT('RC1', 5, 'RC1 PR[5]', status)
END RSTPRCMT
```

38.6.18 RSET_REALREG Built-in ROUTINE

Purpose:

To store a REAL value in the specified register of remote host.

Syntax:

RSET_REALREG (host_port, register_no, real_value, status)
Input/Output parameters
[in] host_port: STRING
[in] register_no: INTEGER
[in] real_value: REAL
[out] status: INTEGER
%ENVIRONMENT RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “register_no” specifies the register into which “real_value” will be stored.
- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sets -12.3 to R [20] of RC1.

```
PROGRAM RSTNRR
%COMMENT = 'RC1 R[20]=-12.3'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status : INTEGER
BEGIN
  RSET_REALREG('RC1', 20, -12.3, status)
END RSTNRR
```

38.6.19 RSET_REG_CMT Built-in ROUTINE

Purpose:

To set comment of numeric register of remote host.

Syntax:

RSET_REG_CMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in] host_port: STRING
[in] register_no: INTEGER
[in] comment_str: STRING
[out] status: INTEGER
%ENVIRONMENT RPCC

Detail:

- “host_port” is IP address or host name of remote robot controller.
- “register_no” specifies which register to set the comments to.
- “comment_str” represents the data which is to be used to set the comment of the given register. If “comment_str” exceeds more than 16 characters, the built-in will truncate the string.
- “status” explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program set comment “RC1 R[15]” to R [15] of host RC1.

```
PROGRAM RSTNRCMT
%COMMENT ='RC1 R[15] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
BEGIN
  RSET_REG_CMT('RC1', 15, 'RC1 R[15]', status)
END RSTNRCMT
```

38.6.20 RSET_SREGCMT Built-in ROUTINE

Purpose:

Sets the comment for the specified string register of remote robot controller.

Syntax:

RSET_SREGCMT (host_port, register_no, comment_str, status)
Input/Output parameters
[in]host_port: STRING
[in]register_no: INTEGER
[in]comment_str : STRING
[out]status: INTEGER
%ENVIRONMENT Group:RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- register_no specifies string register to set.
- comment_str contains the comment to set to the specified string register.
  If the comment_str exceeds more than 16 characters, the built-in will truncate the string.
- status explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sets comment "RC1_STRING_REG_3" to SR[3] of host RC1.

```
PROGRAM RSTSRCMT
%COMMENT ='RC1 SR[3] CMT'
%NOLOCKGROUP
%ENVIRONMENT RPCC
VAR
  status  : INTEGER
BEGIN
  RSET_SREGCMT('RC1', 3, 'RC1_STRING_REG_3', status)
END RSTSRCMT
```

38.6.21 RSET_STR_REG Built-in ROUTINE

Purpose:

To set specified value for the specified string register of remote host.

Syntax:

RSET_STR_REG (host_port, register_no, value, status)
Input/Output parameters
[in] host_port: STRING
[in] register_no: INTEGER
[in] value : STRING
[out] status: INTEGER
%ENVIRONMENT Group: RPCC

Detail:

- "host_port" is IP address or host name of remote robot controller.
- "register_no" specifies string register to set.
- "value" contains the value to set to the specified string register.
- "status" explains the status of the attempted operation. If not equal to 0, then an error occurred.

Example:

Following program sets SR [4] of host RC1.

```
PROGRAM RSTSREG
%COMMENT ='Set RC1 SR[4]'
%NOLOCKGROUP
%ENVIRONMENT rpcc
CONST
  ERR_SUCCESS = 0
VAR
  status    : INTEGER
BEGIN
  RSET_STR_REG('RC1', 4, 'RC1 of SR[4] was set by remote host', status)
END RSTSREG
```

38.7 TIME OUT AND RETRY

When communication doesn’t complete in time because of some kind of problem like network trouble, this function disconnect connection and tries to communicate again. After 2 times retry, this function gives up communication. Standard data transfer program automatically posts "DTBR-002 RPC Call timed out". Status of built-in of this function will be set to corresponding value.

Timeout value is 2 seconds by default. The number of retry is 2. When communication completely failed, this function gives up in about 6 seconds.

Time out value is stored in $DTBR_CFG.$RPC_TIMEOUT in seconds. Minimum timeout value is 1 second regardless of value of the system variable.

If controller cannot recognize host name at all, communication can be given up before time out happens.

Detection of communication error takes more time than usual if timeout value is longer than default.

This function does not give up communication and retry even if program is paused or aborted. If timeout time longer than usual, time you have to wait before starting next communication becomes longer.

38.8 LIMITATIONS

A) Performance of communication is not guaranteed. It depends network traffic and load of CPU of controller.

B) This function affects to performance of other functions that uses Ethernet and vise versa.

C) Request from multiple tasks (programs) doesn’t cause parallel execution of data transfer.
They are actually processed sequentially.

D) In case that the controller to communicate with (server) is R-30*i*Aor R-30*i*A Mate controller. series of the system software of that controller must be 7DA5 or later.

E) You cannot restrict data transfer by the other functions, for example password function.

F) A robot controller cannot send request to itself by this function.

G) Transfer of position register pays attention to the number of axis or extended axis so that data to be set to destination does not exceeds axes destination controller has. However, the other elements of position data are not considered.
For example, followings are not considered.
   - Whether robot can have Cartesian data or not.
   - Axis limit
   - Configuration of position available and range of turn number.
   - Type of axis (rotary or linear)

H) Pause or force abort of program does not stop communication once it started.
Next request should be issued after completion of previous one.

38.9 CAUTION

A) This function can not be used instead of hardwired I/O, field network.
This function offers ability to read/write I/O value of remote controller. However, hardwired I/O and field network I/O is always far faster.

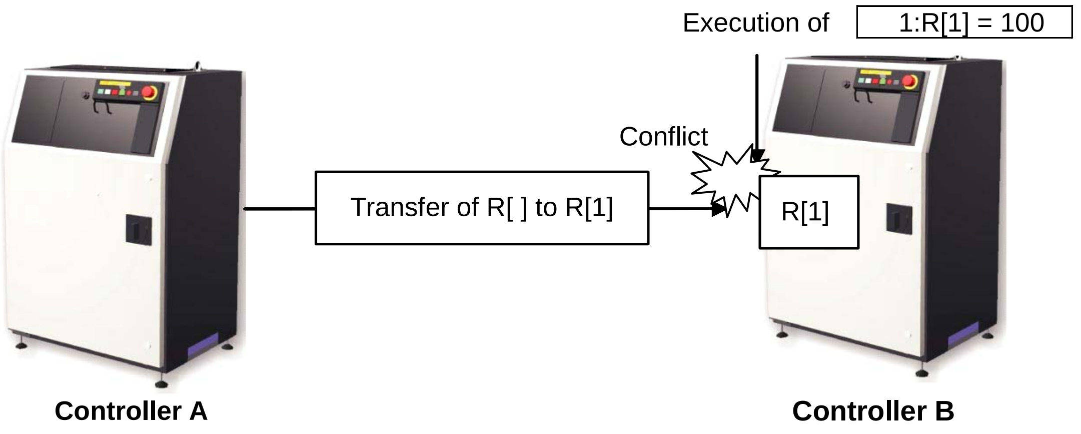
B) Cyclic execution increases load of network and CPU of controller. Consecutive or cyclic execution of data transfer should be avoided.

C) Caution of transfer of position register
Content of position register depends on type of robots and configuration of position.
Please be careful for use of transferred position register.
Especially, you should be very careful when you use transferred position register as destination point of motion statement.
Robot of server and client are not always robot of the same type. Interpretation of position register depends on currently selected tool and user frame.

38.10 CONFLICT BETWEEN DATA WRITES

When you transfer register or position register, please note that only one controller changes a register or position register. This is also true to string register and I/O if you handle them by built-in.

If both server and client write to the same data, they overwrite value written by the other controller. This might look as if data transfer was not working properly.
Especially, if timing of write by client and server is very close, the earlier one may not be recognizable.



When client writes register, position register or string register of server, value and comment is acquired first. Then the client overwrites necessary part of acquired data and request server to write the updated value as a whole.
If server writes slightly before client, value or comment may be changed back to those were got by client.

38.11 TROUBLE SHOOTING

“DTBR-002 RPC Call timed out” is posted.

Please check if the controller (client) can always, stably communicate with specified host (server). You can perform “PING” in PING screen of “Host Comm” setup screen.
If standard data transfer program or built-in of this function are used very often, please lessen frequency of use.

If standard data transfer program or built-in of this function is executed consecutively, there should be interval. If there is already interval, please lengthen it.

Modification of arguments of program is not reflected after resume.

If you modified arguments of standard data transfer program, please move cursor to one line before the CALL and resume the program from the line.

Communication takes time or speed is unstable.

Speed of data transfer by this function is not guaranteed.
Performance depends on network traffic, load of CPU of client and server.
Please improve network traffics.
Please improve timing of data transfer. For example, data transfer that is not needed during motion of robots can be done when robot is not moving.

“INTP-320 (program name, line number) Undefined built-in” is posted.

If this symptom happens when standard data transfer program or built-in of this function is executed, please confirm whether this function is ordered or not. This function is optional.

Time out does not happen. Program keeps running.

Pause or abort of program during data transfer does not stop the transfer once started. It tries to transfer data until the last retry fails.
The next request of data transfer is not processed until previous data transfer completes (fails).

Suppose following procedure happened.

1 Ethernet cable is not plugged in properly.
2 Standard data transfer program, RGETNREG is called but the user aborted the program immediately by FCTN menu.
3 Run the same program again and RGETNREG is called immediately.

In this case, the 2nd RGETNREG times out after about 2 times longer times than usual has passed.
By default setting, it is about 12 seconds.
Once communication is started, waiting for completion or error would be better lather than forcing abort.

Time out error happens just after program resume.

Program pause does not stop data transfer once it started. If timeout occurs while program is paused, timeout error is posted at resume.

Value of axis of joint position of PR[ ] is not transferred properly.

If type of axis (rotary or linear) is different between client’s and server’s corresponding axes, transferred value is not interpreted in the same way. It is not treated as the same value.

“INTP-311 (program name, line number) Uninitialized data is used" is posted when transferred position register is used.

1 Please confirm source data is initialized.
2 The number of total axes or extended axes may be different.
If source data does not have enough axes to set all axis data to destination data, axes that cannot be set because of lack of data becomes un-initialized. In this case, proper data should be set before use.
Suppose group 1 of host RC1 has an extended axis and group 1 of RC2 have no extended axis.
If Cartesian position of group1 of RC2 is set to group1 of RC1, E1 is set to un-initialized value.
RC1 have to set proper value by it self. If 0 is appropriate, PR[1,7]=0 can be used.

Position data of sent/received position register is partially un-initialized.

Please refer to previous item.

Position register screen doesn’t display “R” unless data of all groups are initialized.
Standard transfer program and built-in transfers only specified group.

“VARS-037 Position register is locked” is posted.

Please check if position register is locked on server or client.

“HRTL-047 Address family not supported” is posted.

Please check if specified host is client itself.
Client cannot operate data of client itself by this function.

Data is sent to/received from a controller different from specified one.

If you didn’t cycle power after change of IP address and host name, please cycle power.
If host name includes underscore, don’t use it.

“HRTL-049 Can't assign requested address” is posted.
“HOST-108 Internet address not found” is posted.

Please check if specified host name or IP address is correct.
If you didn’t cycle power after change of IP address and host name, please cycle power.
Please check if the controller (client) can always, stably communicate with specified host (server). You can perform “PING” in PING screen of “Host Comm” setup screen.

39 TOUCH SENSING

Touch sensing allows the robot to change a path automatically to compensate for object displacement. Touch sensing consists of two phases: Search Motion phase and Touch Offset phase.

To use this function, Touch sensor function (A05B-2600-J536) is required. This function was originally developed for ArcTool. Therefore, explanations and figures on this chapter are close to arc welding. But, it is also possible to use this function for HandlingTool.

Search Motion

Search Motion can detect the current position of workpiece.

- Move touch sensor part (on arc welding robot, the top of wire = TCP) toward the workpiece using pre-defined robot motion, speed, and direction.
- Use an input signal to indicate that the robot has come into the contact with the object.
- Store the found location of the workpiece, or position offset information, in position registers.

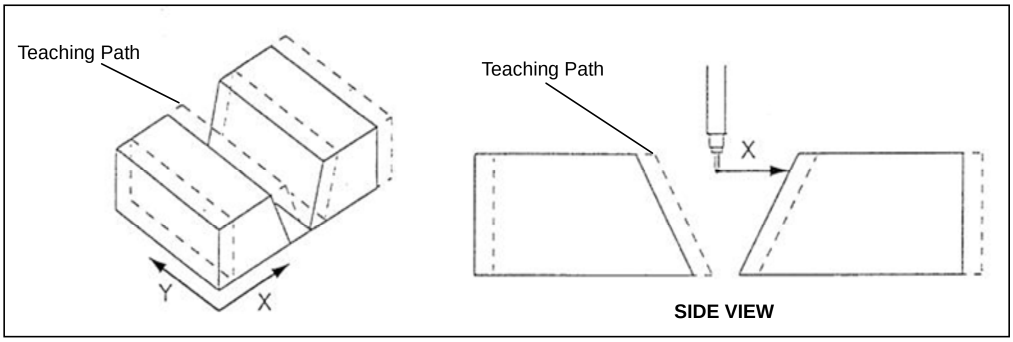


Fig. 39(a) Search motion for detecting shift of X direction

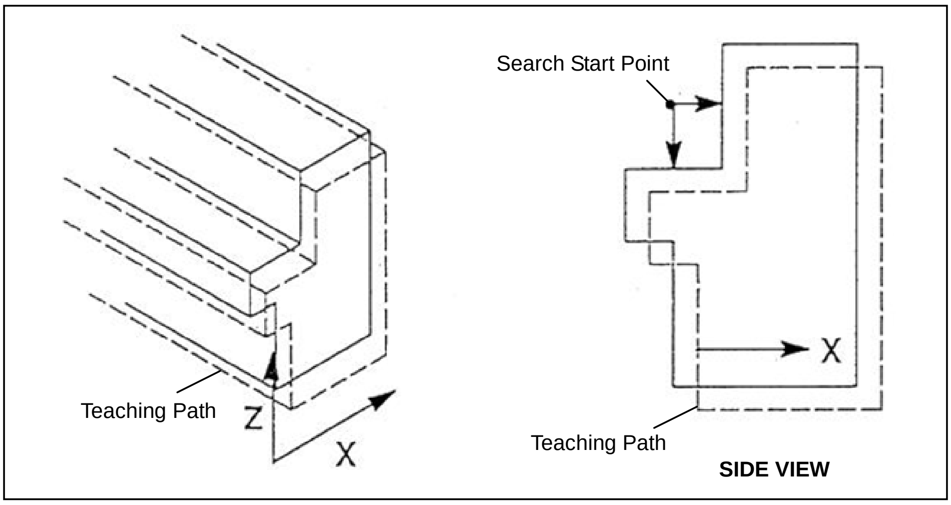


Fig. 39(b) Search motion for detecting shift of X, Z directions

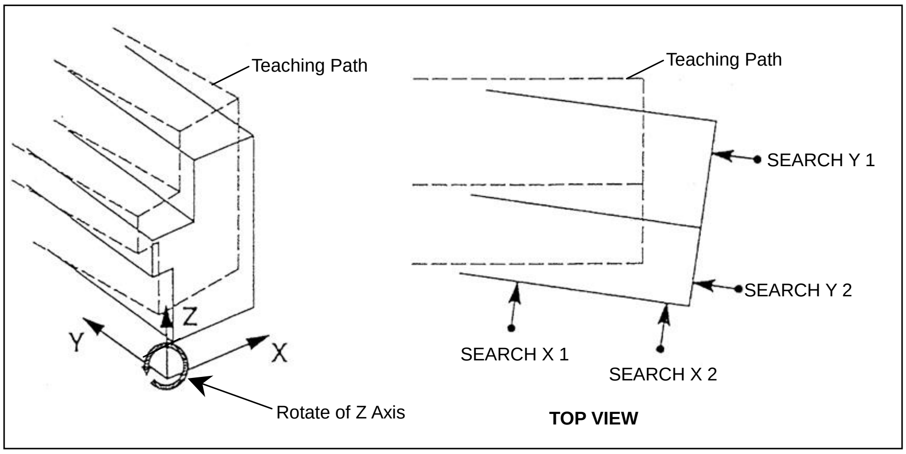


Fig. 39(c) Search motion for detecting shift of X, Y directions and Z rotation

Touch Offset

Shift one or more teaching positions in your welding program by using the stored position the stored position offset information.

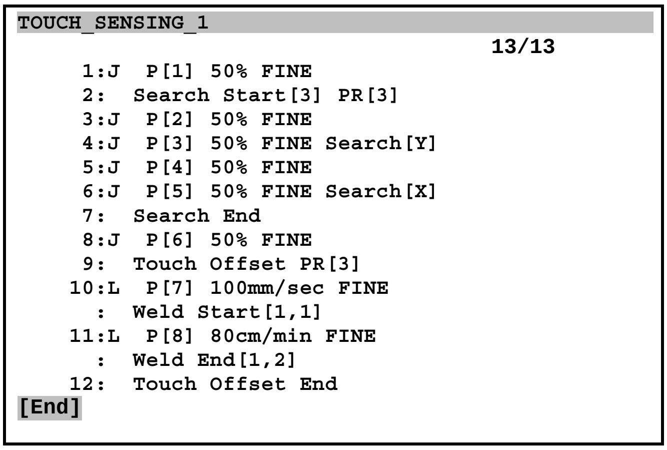
Flow of Touch Sensing

To use touch sensing you must:

- Set up the robot Tool Center Point (TCP) properly.
- Set up touch sensing hardware. The hardware monitors an input signal to determine when the robot comes into contact with the object. There are some weld equipment models which install touch sensing hardware as standard such as Lincoln weld equipment.

After finishing the above preparations, please perform the following setups by referring to this chapter.

- **Assignment of Touch Sensing I/O (Section 39.1)**
  Assign I/O to enable and use the electrical interface circuit.
- **Setup of Touch Sensing Frame (Section 39.2)**
  Create Touch Sensing Frame to decide search direction during search motion.
- **Setup of Touch Sensing Schedule (Section 39.4)**
  Set the method to approach the workpiece (Search Pattern) and define the storing type of obtained position information (absolute position or position offset).
- **Create Touch Sensing Program (Section 39.5)**
  Create touch sensing program by using Search instructions and Touch Offset instructions.
- **Execute Touch Sensing Program (Section 39.6)**
  Execute Search Motion with Master Flag ON and obtain the master position. After that, set Master Flag OFF.



```
TOUCH_SENSING_1
                                          13/13
    1:J  P[1] 50% FINE                              Teach Home Position
    2:  Search Start[3] PR[3]                       Start Search Motion. Use Touch Sch 3, PR[3]
    3:J  P[2] 50% FINE                              Teach a search start position
    4:J  P[3] 50% FINE Search[Y]                    Search to Y direction
    5:J  P[4] 50% FINE                              Teach an another search start position
    6:J  P[5] 50% FINE Search[X]                    Search to X direction
    7:  Search End                                  Finish Search Motion.
    8:J  P[6] 50% FINE                              (If needed) Teach an intermediate point
    9:  Touch Offset PR[3]                          Following points will be offset by PR[3]
   10:L  P[7] 100mm/sec FINE                        P[7] is offset by PR[3]
     :  Weld Start[1,1]
   11:L  P[8] 80cm/min FINE                         P[8] is offset by PR[3]
     :  Weld End[1,2]
   12:  Touch Offset End                            Finish offset of positions
[End]
```

Fig. 39(d) Example program including touch sensing routine

39.1 ASSIGNMENT OF TOUCH SENSING I/O

To use touch sensing you must assign the

- Input signal that the touch sensing circuit monitors to indicate when the robot has reached the object.
- Output signal that enables and disables the touch sensing circuit.

You must wire the necessary connections for the input and output signals to be used for touch sensing. The wire stick detection circuit on the process I/O board also can be used for touch sensing.

Input Signal (Touch Detect Signal)

The touch sensing input signal indicating contact with a part is monitored by the touch sensing circuit. When the input is received, the current robot position is stored in a position register.

RI, DI, WI WS can be used as the touch sensing input signal:

REFERENCE: You can also set up touch sensing to monitor the condition of any RO or DO signal as an input signal. When the selected output turns on during a touch sensing routine, the controller reads this as a received input signal.

Output Signal (Touch Sensing Enable/Disable Command Signal)

RO, DO, WO, WSE can be used to enable the touch sensing circuit:

Assigning the Touch Sensing Inputs and Outputs

You must assign touch sensing inputs and outputs to match the hardware interface at your site. This involves assigning both input and output type and port number. Please refer to Procedure 39-1.

After assignment of Touch Sensing I/O is finished, please check whether touch sensing I/O can properly work or not. Please refer to Procedure 39-2.

> **NOTE**
>
> To use touch sensor circuit on welding power supply for Touch Sensing function, specify the port number of I/O of Touch Sensing signal on Touch sensing setup screen after confirming them on Weld I/O screen.

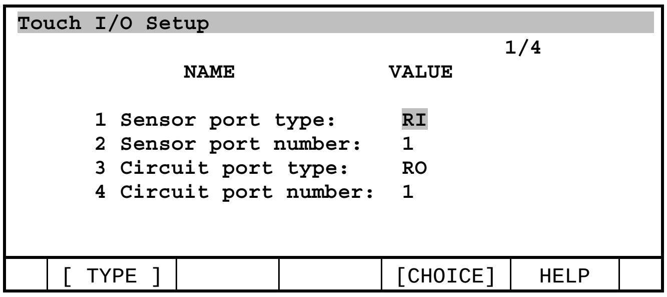
Procedure 39-1 Assigning Touch Sensing Input and Output Signals

Condition

- (Only for multi group system) Motion Group No. is selected for assignment of Touch Sensing I/O.

Step

1 Press [MENU] key.
2 Select SETUP.
3 Press F1, [TYPE] and then select Touch I/O. You will see a screen similar to the followings.



```
Touch I/O Setup
                                      1/4
             NAME             VALUE

      1 Sensor port type:      RI
      2 Sensor port number:    1
      3 Circuit port type:     RO
      4 Circuit port number:   1
```

4 Assign Sensor (input) and Circuit (output) types as follows:
  a Move the cursor to the line you want to assign.
  b Press F4, CHOICE.

c Move the cursor to the desired input/output type.
d Press [ENTER] key.

5 Assign Sensor and Circuit number:
a Move the cursor to the line you want to assign.
b Type the value and press ENTER.

NOTE
This screen shows the settings of the currently selected motion group. To view the settings of another motion group, change the motion group by selecting an auxiliary menu item CHANGE GROUP.

NOTE
After deciding Touch Sensing I/O, please enter the comment which indicates that this signal is used for Touch Sensing. This operation should be performed on each I/O signal screen.

NOTE
The ArcTool software checks the validity of the port type and port number when running your program that includes touch sensing. If the port type or number is invalid, the system displays an I/O invalid error message.

Procedure 39-2 Test for Touch Sensing I/O

Condition

- Workpiece that can be detected by Touch Sensing is placed.
- Assignment of Touch Sensing I/O is finished.

Step

1 Move the robot near the workpiece by jog operation.
2 Display the Output Signal screen for Touch Sensing Output signal set by Procedure 39-1 (If port type is DO, display Digital Output Signal screen). Move the cursor on the line of the port number for Touch Sensing Output signal and then press F4, ON key. The signal is output and Touch Sensing circuit becomes Enabled.
3 Jog the robot and touch the touch sensor part to the workpiece. In this operation, you must set low override value.
4 Display the Input Signal screen for Touch Sensing Detect signal set by Procedure 39-1 (If port type is DI, display Digital Input Signal screen). Next, check the status of port number for Touch Sensing Input signal. If the status is ON, Touch Sensing will correctly work. If the status is OFF even when the touch sensor part touches to the workpiece, please check hardware connection or assignment of Touch Sensing I/O by Procedure 39-1.
5 Move the cursor on the output signal set ON by Step 2 and then press F5, OFF.

⚠WARNING
If Touch Sensing Output signal is set to ON by Step 2, current flow is generated in Touch Sensing Circuit. Therefore, do not touch the touch sensor part (on ArcTool, the top of wire). Additionally, do not forget to return the output signal to OFF.

39.2 SETUP OF TOUCH FRAME

A touch frame determines the motion direction of the robot TCP (touch sensor part). A touch frame is defined by three points. The first point defines the origin, or starting point. The second point defines the positive X direction of the touch frame. The third point defines the positive X-Y plane. Z direction is created by defined X, Y axes.

Fig. 39.3 shows a touch frame and how it is used in a touch sensing program. The orientation of the touch frame to the object is arbitrary in Fig. 39.3. The positive X axis could be aligned with the current z direction. This would re-define positive z to be in the opposite direction of the current positive x direction.

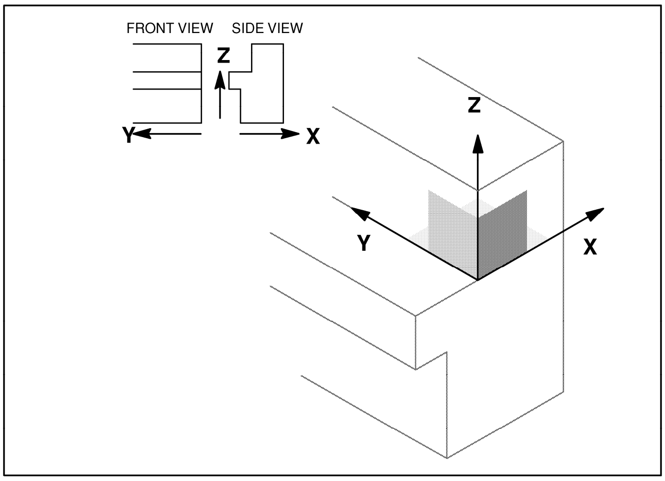


Fig. 39.2 Touch frame

> **NOTE**
> You can set up a maximum of 32 touch frames for each robot group. Touch frames are set up using the touch frame screen in the Setup menu.

> **NOTE**
> You must define a touch frame before you perform a search motion in a program.

There are two ways to define touch frames: The teaching method and the direct entry method. The teaching method defines the touch frame by recording three points. The direct entry method defines the touch frame by the rotation angle value you enter in the touch sense setup screen.

Normally, Reference Group in Touch frame setup screen should be the same group number with Robot Group (Touch frame relates to the WORLD frame of the robot).

In coordinated motion system, if you want to define touch frame to the coordinated (reference) group, you should set leader group number to Reference Group and follower group number to Robot Group.

Table 39.3 lists and describes the items you must set to define the touch frame. About more detail, refer to Section 39.10.

Table 39.2 Touch frame setup items

ITEM	DESCRIPTION
Frame Number	This item specifies the number of the touch frame you want to define.
Reference Group	This item specifies the reference group you want to make the relationship to Touch Frame. Normally, you should set the same group number of below Robot Group item. If you want to perform Coordinated Touch Sensing, setup the item by referring to Section 39.10.
Robot Group	This item specifies a motion group for which the touch frame is used. You cannot select non-robot group (Ex: positioner group). In single group system, always 1 should be set.

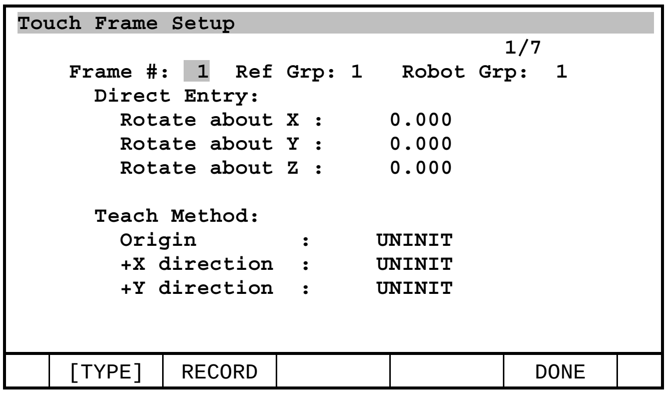
ITEM	DESCRIPTION
Direct Entry (Procedure 39-4)	
Rotate about X	This item specifies the rotation about X for the touch frame.
Rotate about Y	This item specifies the rotation about Y for touch frame.
Rotate about Z	This item specifies the rotation about Z for touch frame.
Teach Method (Procedure 39-3)	
Origin	This item allows you to record the origin of the touch frame.
+X direction	This item allows you to define the +X direction of the touch frame.
+Y direction	This item allows you to define the +Y direction of the touch frame.

Use Procedure 39-3 to define your touch frame by using the teaching method. Use Procedure 39-4 to define your touch frame by using the direct entry method.

Procedure 39-3 Setup of Touch Frame Using Teaching Method

Step

1 Press [MENU] key.
2 Select Setup.
3 Press F1, [TYPE].
4 Select Touch Frame. The following screen will be displayed.



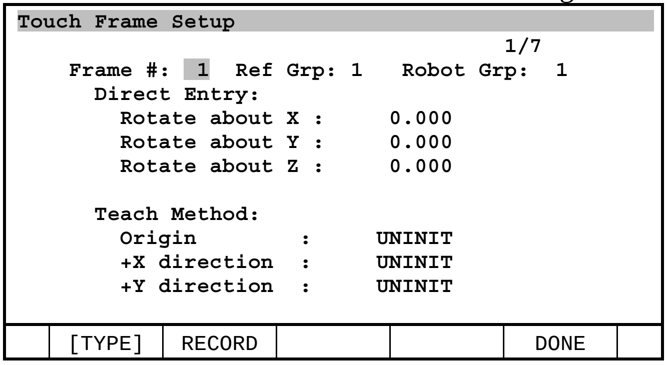
```
Touch Frame Setup
                                             1/7
    Frame #:  1  Ref Grp: 1   Robot Grp:  1
      Direct Entry:
        Rotate about X :     0.000
        Rotate about Y :     0.000
        Rotate about Z :     0.000

      Teach Method:
        Origin        :     UNINIT
        +X direction  :     UNINIT
        +Y direction  :     UNINIT
```

5 Move the cursor to Robot Grp. Enter the robot group number and press [ENTER] key.
6 Move the cursor to Frame Number. Enter the number of the frame to define and press [ENTER] key.
7 Move the cursor to Ref Grp. Normally, enter the group number which is the same as Robot Group item, and then press ENTER. If you want to perform Coordinated Touch Sensing, enter the group number of leader group.
8 Define the origin point of the Touch Frame.
  a Move the cursor to Origin.
  b Jog the Robot TCP to the desired starting point (origin).
  c Press F2, RECORD. (Then, UNINIT will be changed to RECORDED.)
9 Define the +X direction.
  a Move the cursor to +X direction.
  b Jog the robot TCP to a point along the +X axis of the touch frame.
  c Press F2, RECORD.
10 Define the +Y direction.
  a Move the cursor to +Y direction.
  b Jog the robot in the +Y direction of the touch frame, to a point on the X-Y plane.
  c Press F2, RECORD.
11 Press F5, DONE to complete the definition of the frame. (Then, RECORDED will be changed to USED.)

Procedure 39-4 Setup of Touch Frame Using Direct Entry Method

Step

1 Press [MENU] key.
2 Select SETUP.
3 Press F1, [TYPE].
4 Select Touch Frame. You will see a screen similar to the following.



```
Touch Frame Setup
                                        1/7
    Frame #:  1  Ref Grp: 1   Robot Grp:  1
      Direct Entry:
        Rotate about X :     0.000
        Rotate about Y :     0.000
        Rotate about Z :     0.000

      Teach Method:
        Origin        :     UNINIT
        +X direction  :     UNINIT
        +Y direction  :     UNINIT
```

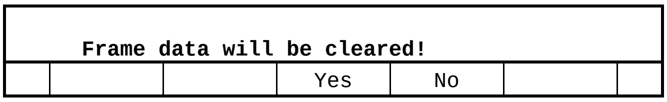
5 Move the cursor to Robot Grp. Enter the robot group number and press ENTER.
6 Move the cursor to Frame Number. Enter the number of the frame to define and press ENTER.
7 Move the cursor to Ref Grp. Normally, enter the group number which is the same as Robot Group item, and then press ENTER. If you want to perform Coordinated Touch Sensing, enter the group number of leader group.
8 Define the rotation angle about X.
 a Move the cursor to Rotate about X.
 b Enter the value (in degrees).
9 Define the rotation angle about Y.
 a Move the cursor to Rotate about Y.
 b Enter the value (in degrees).
10 Define the rotation angle about Z.
 a Move the cursor to Rotate about Z.
 b Enter the value (in degrees).
11 Press F5, DONE to complete the definition of the frame. (UNINIT will be changed to USED.)

⚠**CAUTION**

Do not forget DONE operation at the last timing of Procedure 39-3 and 39-4. If you forget this operation, touch frame can not be defined.

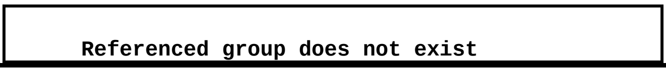
NOTE

If you change the value of Ref Grp for an initialized frame at Step 7 of Procedure 39-3 and 39-4, the following warning message will be displayed:



```
Frame data will be cleared!
```

If you press F3, Yes, the frame data will be reinitialized.
If you set different number from Robot Group number to Ref Grp, but no coordinated pair calibration is not finished for the group number, the value of Ref Grp will return to the original value and the following warning message will be displayed:



```
Referenced group does not exist
```

39.3 SEARCH PATTERN

Search Pattern and Pattern Type determine the offset direction by Touch Sensing function. You should select Search Pattern and Pattern Type after checking the expected direction for shift of workpiece.
For detecting horizontal shift of workpiece, one search motion per 1 search direction is necessary. If the detection of rotational shift of workpiece is also requested, two search motions per 1 search direction are necessary.

Search Pattern

Four types of search patterns are available. You should select one of following search patterns by the shape of workpiece.

- Simple search
- Fillet/lap search
- V-Groove search
- Outside/inside diameter search

By selected search pattern, the position data type stored to a position register is defined. The type becomes Absolute position or offset value.
If you select Fillet/lap search, you should also select search type by the possibility of shift directions of workpiece.

Simple Search

For a simple search, a two-dimensional search is executed to find the actual location of one position on a workpiece. A simple search stores the found absolute position (x, y, z, w, p, r) into a position register PR[ ]. Once completed, the robot is programmed to move to the position stored in that position register.

Simple search requires:

- Surfaces being searched are perpendicular to each other.
- Searches to be done in two different directions.
- The second search motion to be performed with the desired torch angle.

The first search defines the positional information for that search direction only (x, for example). The second search defines the other direction positional information (z, for example). The starting position of the second search defines the remaining positional information, (y, w, p, r, for example) that determines the torch angle for welding and, in this case, the Y value.

Simple search is typically used to find the starting point of a weld path that uses the Thru-Arc Seam Tracking (TAST) option or Automatic Voltage Control (AVC) Tracking option.
A two-dimensional search is programmed in the software as the only valid search pattern type when a simple search is used. Changing the search pattern type has no effect.

> ⚠**WARNING**
> Do not use simple search when you use the multipass option with touch sensing because both simple search and multipass use position registers. Simple search stores the computed position in a position register. Multipass cannot use position registers to plan paths. Use the 2D fillet search pattern when using multipass with touch sensing.

See Fig. 39.3(a) for an illustration of a simple search routine. Refer to Section 39.5 for example programs using simple search.

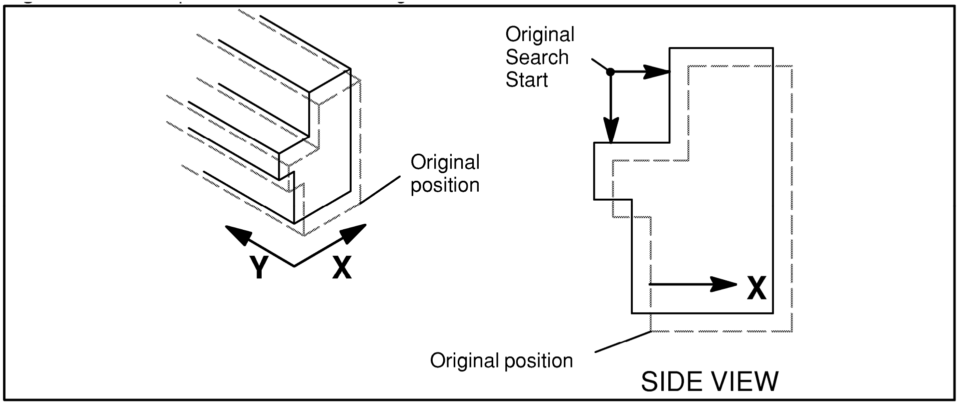


Fig. 39.3 (a) Simple search routine using searches in two directions

Fillet/Lap Search

For Fillet/Lap Search a one, two, or three-dimensional searches are executed to obtain positional offset information. Fillet/Lap Search stores positional offset information in a positional register PR[ ]. This offset can be applied to one or more positions in a programmed path.

1D Shift, 2D Shift, 3D Shift, 1D + Rot, 2D + Rot, 3D + Rot can be used as Search Pattern Type for Fillet/Lap Search. You can offset for on, two or three dimensions. Additionally, rotational offset for compensation can be also used.

For example, when 1D + Rot is selected as Search Pattern Type, the fillet search can offset for a rotation about Z by searching 2 points to X direction. Note that is this type of search, the first touch point is used as the arc start point. See Fig. 39.3(b).

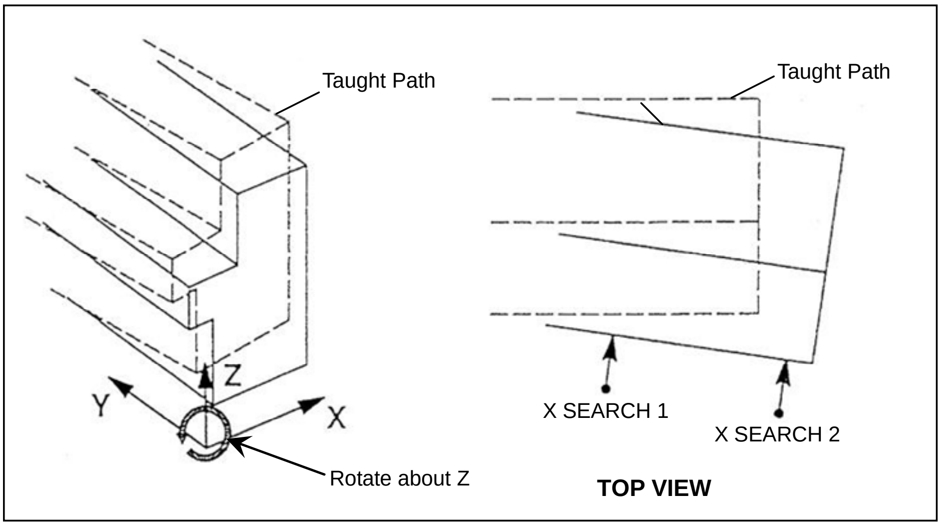


Fig. 39.3 (b) Fillet search in one direction (x) with rotation about z

When 2D + Rot is selected as Search Pattern Type, the fillet search can offset for a rotation about Z by searching 2 points per one direction to X and Y directions. See Fig. 39.3(c).

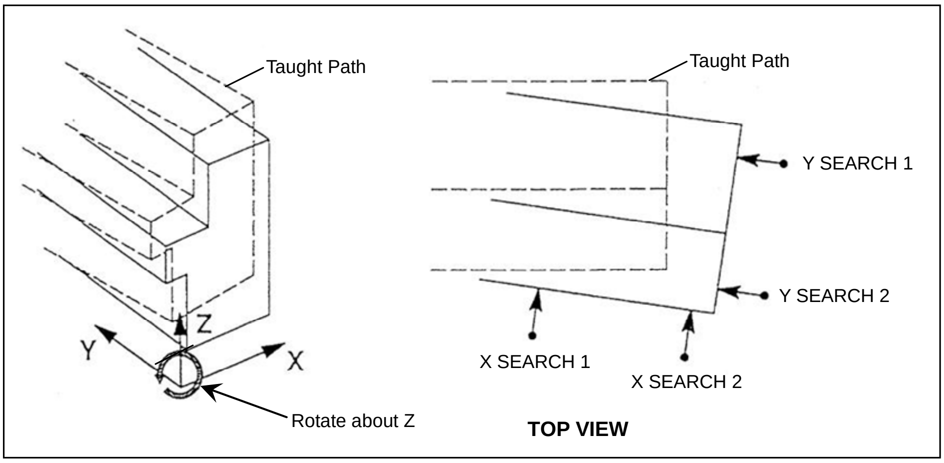


Fig. 39.3 (c) Fillet search in two directions (X and Y) with rotation about Z

When 3D + Rot is selected as Search Pattern Type, the fillet search can offset for rotations about all directions by searching 3 points to another direction in addition to 2D + Rot search motion. See Fig. 39.3(d).

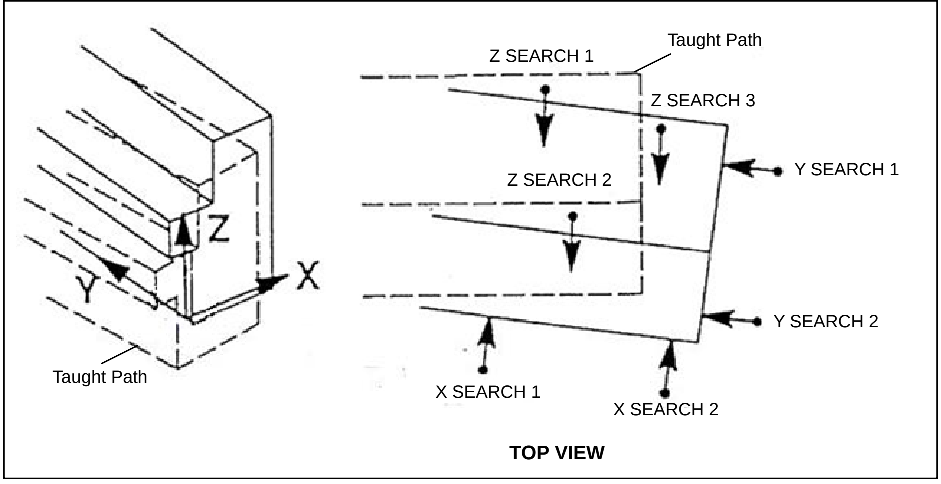


Fig. 39.3 (d) Fillet search in three directions (X, Y, Z) with rotation about all directions

Refer to Section 39.5 for example programs using Fillet/Lap Search.

V-Groove Search

For V-Groove Search a one-dimensional search is executed to obtain positional offset information. A V-Groove Search stores positional offset information in a positional register [PR]. This offset can be applied to one or more positions in a programmed path. Only 1D Search Pattern Type is allowed for V-Groove Search.

See Fig. 39.3(e) for an illustration of a V-Groove Search. Refer to Section 39.6 for example programs using V-Groove Search.

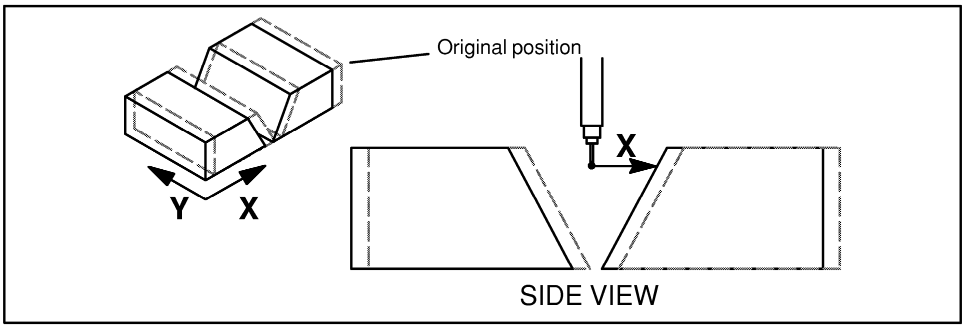


Fig. 39.3 (e) V-groove search

Outside/Inside Diameter Search (OD/ID)

For Outside/Inside Diameter Search (OD/ID Search) a two-dimensional search is executed to obtain the positional offset information of the center point of a circular path relative to the original (master) location. An Outside/Inside Diameter Search stores positional offset information in a positional register [PR]. This offset can be applied to one or more positions in a programmed path.
Only 2D Search Pattern Type is allowed for OD/ID Search. About more detail, refer to Table 39.4.

See Fig. 39.3 (f) for an illustration of a OD/ID Search.

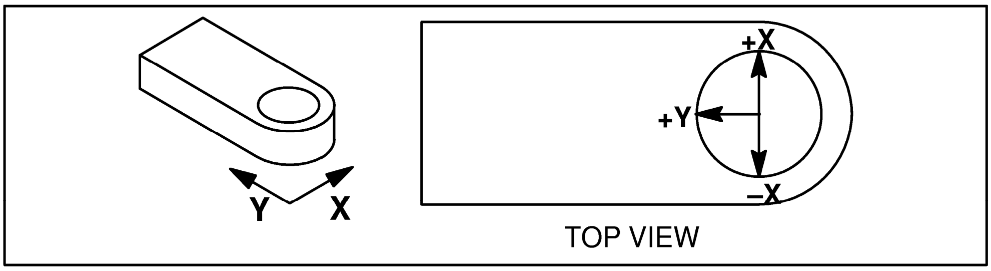


Fig. 39.3 (f) OD/ID search in two directions (x and y)

Table 39.4 shows a matrix of possible search pattern and valid pattern types. Select a combination that you would like to use on your application and verify that it will provide the proper results.

Table 39.3 Search pattern and valid pattern type

Search Patterns	Pattern Type 1_D	Pattern Type 2_D	Pattern Type 3_D	Pattern Type 1_D and Rotation	Pattern Type 2_D and Rotation	Pattern Type 3_D and Rotation
Simple Search	Not Valid	Requires 2 different search directions. 1 search per direction.	Not Valid	Not Valid	Not Valid	Not Valid
Fillet/Lap	Requires 1 search direction. 1 search.	Requires 2 different search directions, x and y, x and z, y and z. 1 search per direction.	Requires 3 different search directions, x, y, and z. 1 search per direction.	Requires 1 search direction. 2 searches per direction.	Requires 2 different search directions. 2 searches per direction.	Requires 3 different search directions. 3 searches in one direction (usually -z) 2 searches in each of the remaining directions.
V-Groove	Requires 1 search direction. 1 search	Not Valid	Not Valid	Not Valid	Not Valid	Not Valid
OD/ID	Not Valid	Requires 3 different searches in 2 different directions. For example, +x, -x, +y, NOT x, y, z. 1 search per direction.	Not Valid	Not Valid	Not Valid	Not Valid

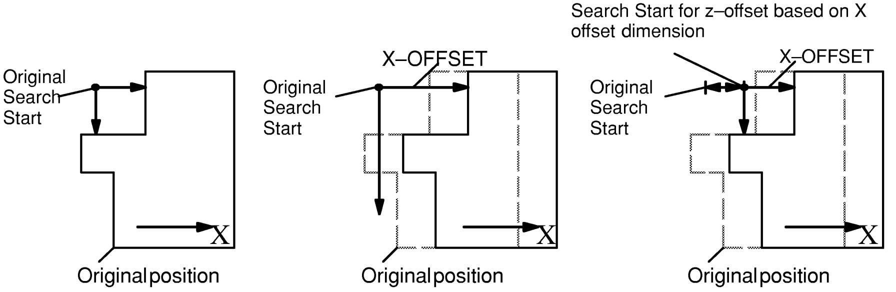
39.4 TOUCH SCHEDULE

A touch schedule is a series of conditions that control how the search motion is completed. In this schedule, Touch Frame, Search Pattern and Pattern Type are included.

Thirty-two touch schedules are available. You can access touch schedules from the DATA menu. There are two screens associated with touch schedules: the SCHEDULE screen and the DETAIL screen.
The SCHEDULE screen allows you to view and set limited information for all schedules at once. DETAIL allows you to view and set the complete information for a single schedule.
Table 39.4 lists and describes each DETAIL screen condition. The items of SCHEDULE screen are also included in touch sensing DETAIL screen.
Use Procedure 39-5 to define touch schedules.

Table 39.4 Touch schedule screen Items

ITEM	DESCRIPTION
Touch Schedule	This item indicates the number of the displayed schedule. A comment can be entered.
Master Flag Default=OFF	This item enables the search routine to be used as a mastering routine for those touch sensing programs that generate positional offset information. If set to ON, when the search routine is executed, the touched positions are recorded on TP program internally as the reference positions to be used by future searches. This flag must be set to OFF after the master search is completed in order to generate the information of position offset on the objects to be searched. When the search is performed with Master Flag ON, the information of position offset in the position register is set to all zero values. The Master Flag condition has no effect on simple search pattern.
Search Speed Default = 50.0 mm/sec	This item specifies how fast the robot will move when performing a Search Motion. The accuracy of search will be improved when this speed is slow. ⚠ **CAUTION** A search motion is programmed as a motion option at the end of a motion instruction. The speed at which the robot will move is determined by the search speed, not by what is indicated in the motion instruction. During testing, when dry run is in effect, this search speed is also used. The dry run speed has no effect.
Search Distance Default = 100 mm	This item defines how far the robot can move when it is performing a search. Error code “THSR-017 No contact with part.” is posted when this distance is reached without making contact with the workpiece.
Touch Frame Default = 1	This item defines the touch frame number to be used in the touch schedule. This determines the X, Y, and Z directions for the search motion. The same touch frame can be used in more than one touch schedule.
Search Pattern Default = SIMPLE	This item defines the type of object to be searched and causes the ArcTool software to compute the found position or positional offset information dependent on the search pattern selected. The computed data is stored in a position register. There are four available search patterns: - Simple Search - Fillet/Lap Search - V-Groove Search - OD/I D Search About more detail, refer to the Section 39.4.

ITEM	DESCRIPTION
Pattern Type Default = 1_D Shift • 1_D Shift • 2_D Shift • 3_D Shift • 1_D + Rot • 2_D + Rot • 3_D + Rot	This item selects the type of offset to be stored in the position register with Fillet/Lap Search Pattern. Six pattern types are available: Stores a one-dimensional offset. Offsets can be in the X, Y, or Z direction. Stores a two-dimensional offset. Offsets can be in two of the X, Y, or Z direction. Stores a three-dimensional offset to a program. Offsets are in the all of X, Y, and Z directions. Stores a one-dimensional offset with rotation about the axis of which the search is not performed. Stores a two-dimensional offset with rotation about the axis of which no searches are performed. For example, if the workpiece is being searched for an offset in both the X and Y directions, a 2_D Shift & Rotate search can offset for a rotation about the z axis. Stores a three-dimensional offset with rotation about the all axes are performed. NOTE: Simple, OD/ID, and V-Groove search patterns are pre-defined. Changing the pattern type for these searches has no effect. See Table 39.4 for valid pattern types for selected search patterns.
Incremental Search Default = ON	Offsets the starting position of the second etc. search in a search routine by the amount of offset found by the first search motion. If set to OFF, the robot returns to the original starting position. The following illustration shows how the incremental search affects the search routine. Incremental search requires a different position number for each search. Program Example: J P[4] 100% FINE J P[5] 100% FINE SEARCH [-X] J P[6] 100% FINE J P[7] 100% FINE SEARCH [-Z]  Search Start for z–offset based on X offset dimension Original Search Start X Original position X–OFFSET Original Search Start X Original position X–OFFSET Original Search Start X Original position Without incremental search, the robot found the X-offset but cannot find the Z-offset. **NOTE** Simple search does not support incremental search.
Auto Return Default = ON	This item moves the robot back to the search start position when contact is made with the object. If set to OFF, the robot stops at the contact point and moves straight to the next position.
Return Speed Default = 100 mm/sec	This item specifies the speed at which the robot will return to the search start position upon making contact with the part.
Return Distance Default = 2000 mm	When Auto Return is set to ON, Return Distance specifies the distance the robot will return automatically. If the return distance passes the initial search start position, the robot will return to the initial start position.

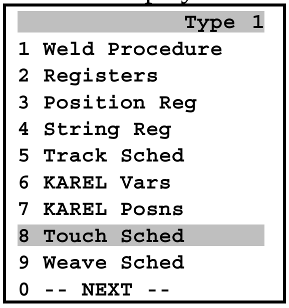
ITEM	DESCRIPTION
Reference Group	This item specifies how the offset is recorded. When Search Pattern is not Simple Search, you should specify the same group number as Ref Grp on selected Touch Frame. When Search Pattern is Simple Search, you should specify the same group number as Robot Group item on this schedule. **NOTE** When search pattern is not simple search, if Reference Group is not equal to Ref Grp on Touch Frame, an error message "THSR-039 Reference grp mismatch," will be posted. When search pattern is simple search, if Reference Group is not the same as the specified number of Robot Group on this touch schedule, an error message, "THSR-037 Illegal motion ref. grp," will be posted.
Return Term Type Default = FINE	This item specifies the termination type the robot will use to return to the search start position. Four Return Term Types are available: - FINE - CNT20 - CNT40 - CNT100
Contact Record PR Default = 32	The search output position register is used as a temporary buffer to hold the last search contact position. The purpose for this temporary position register buffer is to provide the ability to look at the positional data of an individual search, or to extract data from the buffer in a program. By default, this register is position register 32. ⚠ **CAUTION** The data in the position register is overwritten at each search motion so the same position register should not be used to store the final positional data from the search motion. Also, the contents of this temporary buffer is a real position, not an offset. Do not program motion instructions to use this position register data as an offset.
Error on Failure Default = ON	This item posts error code “THSR - 017 No contact with part”, if the search move exceeds the distance set in Search Distance. When this item is OFF, the program execution continues with the next instruction even if the Search Distance is exceeded. Programming Hint: If this is set to OFF, the next instruction in the program looks at the contents of the Error Register and branch accordingly.
Error Register Number Default = 32	When Error On Failure is set to OFF, this register is set to 1 when the search distance is exceeded. A successful search sets this register to 0.
Robot Group Default = [1,*,*,*,*,*,*,*]	This item specifies the robot group that uses the touch sensing schedule.

Procedure 39-5 Setup of Touch Schedule

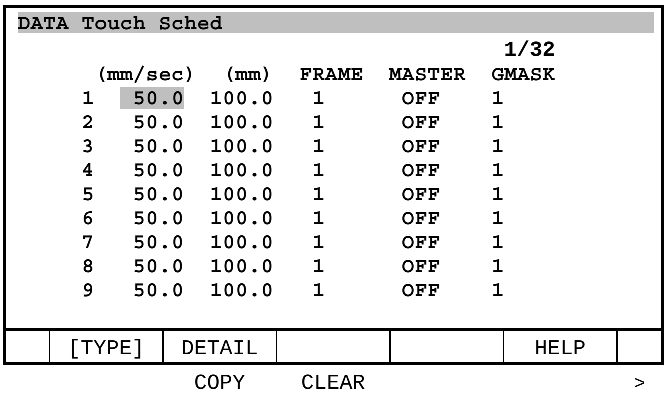
Step

1 Press [DATA] key.

2 Press F1, [TYPE]. Following menu will be displayed.



3 Select Touch Sched. You will see a screen similar to the following.



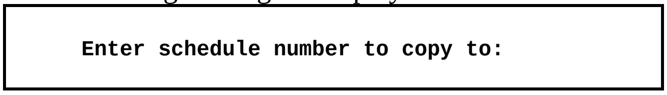
```
DATA Touch Sched
                                            1/32
      (mm/sec)  (mm)   FRAME  MASTER  GMASK
     1    50.0  100.0    1      OFF     1
     2    50.0  100.0    1      OFF     1
     3    50.0  100.0    1      OFF     1
     4    50.0  100.0    1      OFF     1
     5    50.0  100.0    1      OFF     1
     6    50.0  100.0    1      OFF     1
     7    50.0  100.0    1      OFF     1
     8    50.0  100.0    1      OFF     1
     9    50.0  100.0    1      OFF     1

  [TYPE]   DETAIL                    HELP
           COPY     CLEAR                      >
```

4 To copy schedule information from one schedule to another:

a Press NEXT key.

b Move the cursor to the schedule you want to copy.

c Press F2, COPY. Following message is displayed.



```
     Enter schedule number to copy to:
```

d Enter the schedule number to which you want to copy the data.

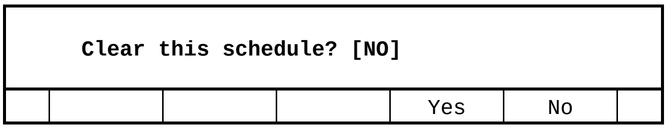
e Press ENTER. The data will be copied, but the comment will not be copied.

5 To clear the information you have entered for a schedule:

a Move the cursor to the schedule.

b Press NEXT key.

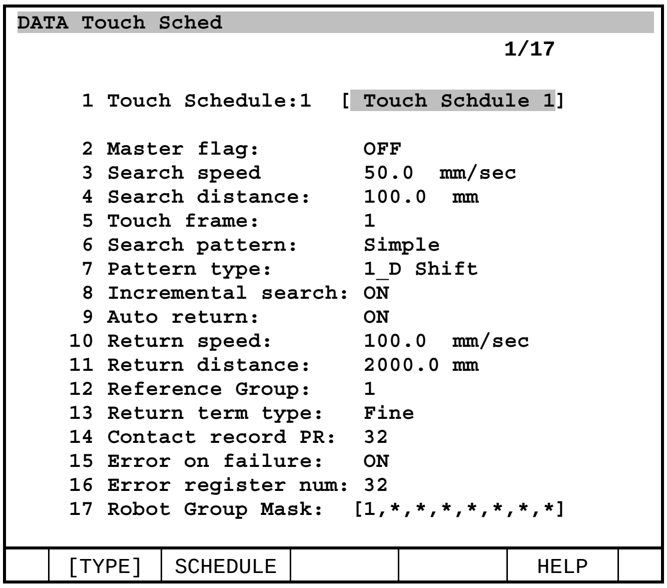
c Press F3, CLEAR. Following message is displayed. If F4, Yes is pressed, the data will be cleared, but the comment will remain.



```
     Clear this schedule? [NO]

                               Yes      No
```

6 Move the cursor to the desired schedule number.

7 To display more information about the schedule, press F2, DETAIL. See the following screen for an example.



```
DATA Touch Sched
                                            1/17

     1 Touch Schedule:1  [ Touch Schdule 1]

     2 Master flag:        OFF
     3 Search speed        50.0  mm/sec
     4 Search distance:    100.0  mm
     5 Touch frame:        1
     6 Search pattern:     Simple
     7 Pattern type:       1_D Shift
     8 Incremental search: ON
     9 Auto return:        ON
    10 Return speed:       100.0  mm/sec
    11 Return distance:    2000.0 mm
    12 Reference Group:    1
    13 Return term type:   Fine
    14 Contact record PR:  32
    15 Error on failure:   ON
    16 Error register num: 32
    17 Robot Group Mask:  [1,*,*,*,*,*,*,*]

  [TYPE]  SCHEDULE                   HELP
```

8 Set each schedule item as desired.
9 To add a comment:
  a Move the cursor to the to the comment line and press ENTER.
  b Select a method of naming the comment.
  c Press the appropriate function keys to add the comment.
  d When you are finished, press ENTER.
10 To select [Search pattern], [Pattern type], [Return Term Type]:
  a Move the cursor to each item and press F4, SELECT.
  b Select the item and press ENTER.

NOTE
Touch Schedule is only for defining method of Touch Sensing, and information of master positions is never saved on Touch Schedule (the information is saved on TP program). Therefore, you can use one Touch Schedule in multiple programs and multiple places.

39.5 TOUCH SENSING PROGRAMMING

A touch sensing routine consists of search instructions to locate an object, and offset instructions to displace programmed positions.

⚠**WARNING**
Recorded positions and position registers are affected by UFRAME, and UFRAME has an affect during playback. If you change UFRAME, any recorded positions and position registers will also change.

NOTE
Any changes to the tool frame affects the search start position.

Touch Sensing Instructions

Following touch sensing instructions are provided:
- Search Start [ ] PR [ ] (Search Start instruction)
- Search End (Search End Instruction)
- Touch Offset PR[ ] (Touch Offset Start instruction)
- Touch Offset End (Touch Offset End instruction)

Additionally, following instruction is used as additional motion instruction between Search Start – Search End instructions.
- Search [ ] (Search instruction)

⚠**CAUTION**
Do not use Replace in (EDCMD) menu on program edit screen to the program that has Touch Sensing instructions. This will cause memory write error. If you want to replace motion instructions, insert Touch Sensing instructions after deleting the motion instruction that you want to replace.

Search Start/End Instructions

Search Start instruction commands the start of search motion. Specify Touch Schedule number for this search motion, and also specify Position Register number for storing positional information by Touch Sensing.

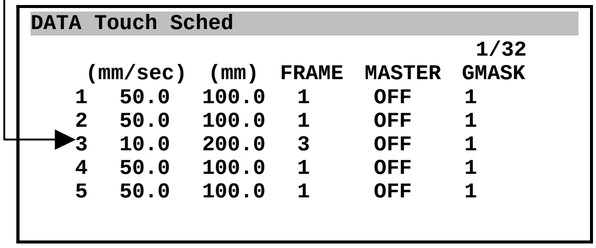
Search Start [i] PR[x]

Touch Schedule Number — Position Register Number

Example
```
1: Search Start[1] PR[2]
 2: Search Start[ R[3] ] PR[ R[5] ]
```

Search Start [3] PR[3] Touch Schedule List screen

Touch Schedule Number	
Search Speed	10.0mm/sec
Search Distance	200.0mm
Touch Frame	3
Master Flag	OFF
Group Mask	1



```
DATA Touch Sched
                                    1/32
    (mm/sec)  (mm)  FRAME  MASTER  GMASK
   1   50.0   100.0   1      OFF     1
   2   50.0   100.0   1      OFF     1
   3   10.0   200.0   3      OFF     1
   4   50.0   100.0   1      OFF     1
   5   50.0   100.0   1      OFF     1
```

Fig. 39.5(a) Search start instruction

One Search End instruction is required for one Search Start instruction. If you want to execute Search Start instruction again for new search motion, you must execute Search End instruction for finishing previous search motion.

Search End

Example
```
1: Search End
```

Fig. 39.5(b) Search end instruction

Search instruction must be taught between this Search Start instruction and Search End instruction.

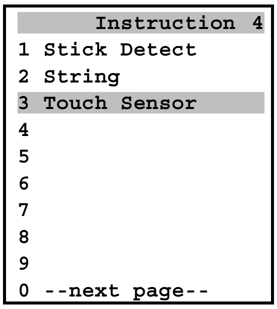
Procedure 39-6 Teach Search Start/End Instructions

Condition

- Program edit screen has been displayed.
- Teach pendant is enabled.

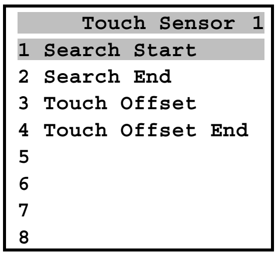
Step

1 Press [NEXT] key and press F1, INST. Following menu will be displayed.



```
       Instruction  4
1 Stick Detect
2 String
3 Touch Sensor
4
5
6
7
8
9
0 --next page--
```

2 Select Touch Sensor. List of touch sensing instructions are displayed. If you teach Search Start instruction, select Search Start.



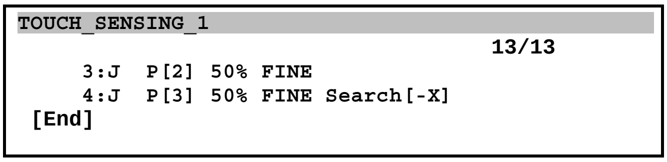
```
    Touch Sensor  1
1 Search Start
2 Search End
3 Touch Offset
4 Touch Offset End
5
6
7
8
```

3 Enter Touch Schedule number to [] just after Search Start. Additionally, enter position register number to [] of PR.
4 If you teach Search End instruction, select Search End on Step 2.

Search Instruction

There is one Touch Sensing motion option: Search [ ]..The Search [ ] motion option directs the motion of the robot (in a positive or negative X,Y or Z direction) to search for the object. The X, Y and Z vectors are defined by the touch frame assigned in the touch schedule.

The taught position in motion instruction that has the search instruction does not has actual meaning, so motion to the search start position must be taught by a separate motion instruction. See Fig. 39.6(c). Additionally, motion speed in motion instruction that has the search instruction does not have actual meaning, either. Instead of this, Search Speed in Touch Schedule (refer to Table 39.5) is used as motion speed. When the touch sensor part touches the workpiece during Search Motion, current position is recorded to Contact Record PR.



```
TOUCH_SENSING_1
                                              13/13
     3:J  P[2]  50% FINE
     4:J  P[3]  50% FINE Search[-X]
 [End]
```

Move to Search Start Position
Search Motion (P[3] is never used)

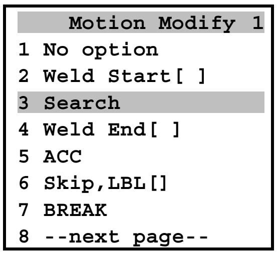
Fig. 39.5(c) Program example using search instruction as additional motion instruction

About how to teach Search instruction, refer to Procedure 39-7.

Procedure 39-7 Teach Search Instruction

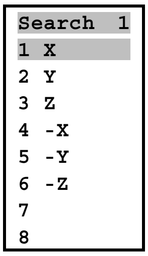
Step

1 Jog the robot to the search start position and teach motion instruction.
2 Teach another motion instruction again at the same position.
3 Move the cursor to the end of the motion instruction taught on Step 2, and press F4, [CHOICE] to view the motion option choices.



```
   Motion Modify  1
1 No option
2 Weld Start[ ]
3 Search
4 Weld End[ ]
5 ACC
6 Skip,LBL[]
7 BREAK
8 --next page--
```

4 Select Search.
5 Select the direction of the search to be performed and press [ENTER] key.



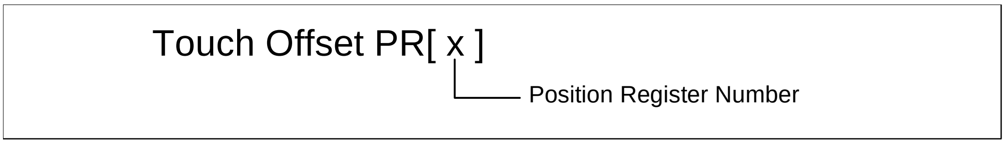
```
Search  1
1 X
2 Y
3 Z
4 -X
5 -Y
6 -Z
7
8
```

⚠CAUTION

Teach Search instruction at the last position of additional motion instruction. Additionally, motion instruction for moving to search start position (motion instruction which is taught just before motion instruction which has Search instruction) must use FINE termination type. If CNT termination type is used for their instructions, correct offset information cannot be calculated.

Touch Offset Start/End Instructions

Touch Offset Start/End instructions are used for shifting positions on the program by using position offset which is obtained by search motion. The position offset by Touch Offset is defined by the position register specified in Touch Offset Start instruction. Normally, specify position register number which record position offset by Search Start/End instructions.



Example

```
1: Touch Offset PR[3]
2: Touch Offset PR[ R[5] ]
```

Fig. 39.5(d) Touch offset start instruction

Touch Offset End

Example

```
1: Touch Offset End
```

Fig. 39.5(e) Touch offset end instruction

About how to teach Search instruction, refer to Procedure 39-8.

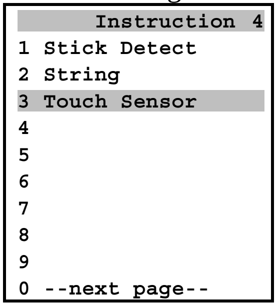
Procedure 39-8 Teach Touch Offset Start/End Instructions

Condition

- Program edit screen has been displayed.
- Teach pendant is enabled.

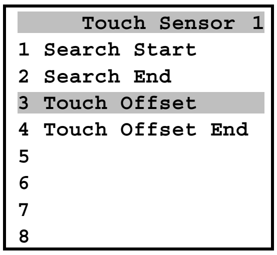
Step

1 Press [NEXT] key and press F1, INST. Following menu will be displayed.



```
      Instruction  4
1 Stick Detect
2 String
3 Touch Sensor
4
5
6
7
8
9
0 --next page--
```

2 Select Touch Sensor. List of touch sensing instructions are displayed. If you teach Touch Offset Start instruction, select Touch Offset.



```
     Touch Sensor 1
1 Search Start
2 Search End
3 Touch Offset
4 Touch Offset End
5
6
7
8
```

3 Enter position register number to [] of PR.
4 If you teach Touch Offset End instruction, select Touch Offset End on Step 2.

> ⚠CAUTION
> Do not use CALL instruction between Touch Offset Start – End instructions. On the motion instructions after the execution of CALL instruction, Touch Offset becomes invalid.

Program Example of Simple Search

After a Simple Search routine, the touch sense software will calculate an absolute (real) position (X,Y,Z,W,P,R) and put the data in the position register defined by the Search Start instruction.

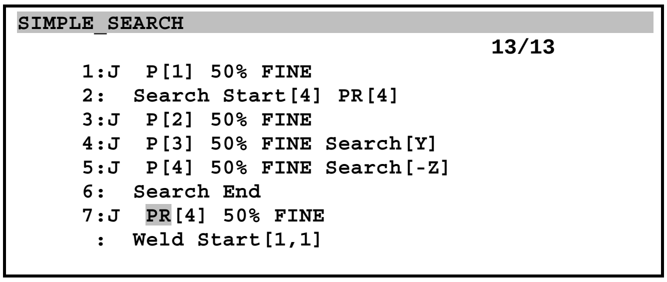
Since this is a real position, the robot will be commanded to move to the position in the position register.

Example: J PR [4] 100% FINE Weld Start [1,1]

Above example shows where position register 4 is the position register specified in the simple search routine.

Simple Search has different points from other Search Patterns.

- Master Flag is always OFF.
- Position values stored in position register is absolute position data.
- Touch Offset Start/End instructions are never required.



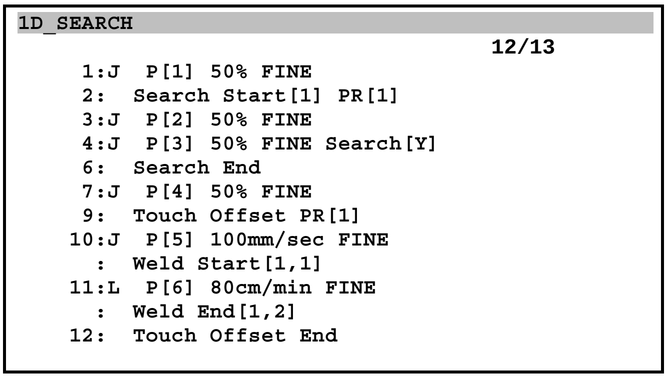
```
SIMPLE_SEARCH
                                            13/13
     1:J  P[1] 50% FINE
     2:  Search Start[4] PR[4]
     3:J  P[2] 50% FINE
     4:J  P[3] 50% FINE Search[Y]
     5:J  P[4] 50% FINE Search[-Z]
     6:  Search End
     7:J  PR[4] 50% FINE
      :  Weld Start[1,1]
```

Teach Home Position
Start Search Motion. Use Touch Sch 4, PR[4]
Teach a search start position
Search to Y direction
Search to -Z direction
Finish Search Motion
Move the robot to PR[4] position,
And start Arc Welding

Fig. 39.5(f) Program example of simple search (Refer to Fig. 39.4(a))

Program Example of 1D Search

In Touch Sensing program except simple search pattern, use the offset information by search motion on Touch Offset instruction, and offset the taught path.
In 1D Search, one point to one direction to which the workpiece will shift is searched, and taught path is compensated by Touch Offset instruction. Refer to the following program example.

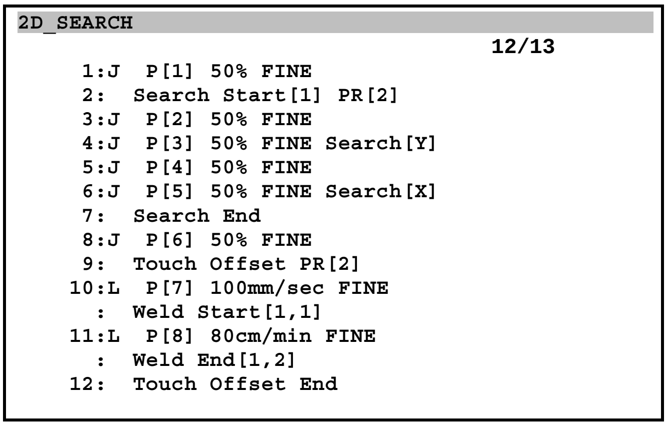


```
1D_SEARCH
                                    12/13
     1:J  P[1] 50% FINE                      Teach Home Position
     2:  Search Start[1] PR[1]               Start Search Motion. Use Touch Sch 1, PR[1]
     3:J  P[2] 50% FINE                      Teach a search start position
     4:J  P[3] 50% FINE Search[Y]            Search to Y direction
     6:  Search End                          Finish Search Motion
     7:J  P[4] 50% FINE                      (If needed) Teach an intermediate point
     9:  Touch Offset PR[1]                  Following points will be offset by PR[1]
    10:J  P[5] 100mm/sec FINE                P[5] is offset by PR[1]
      :  Weld Start[1,1]
    11:L  P[6] 80cm/min FINE                 P[6] is offset by PR[1]
      :  Weld End[1,2]
    12:  Touch Offset End                    Finish offset of positions
```

Fig. 39.5(g) Program example of 1D search (Fillet/Lap search and V-Groove search)

Program Example of 2D Search

In 2D Search, search to two directions are performed, and the search point is one point per one direction. Then taught path is compensated by Touch Offset instruction. Refer to the following program example.

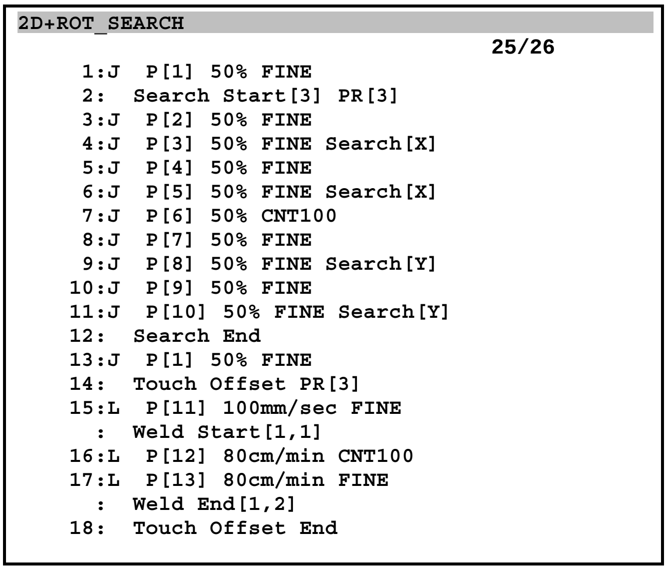


```
2D_SEARCH
                                    12/13
     1:J  P[1] 50% FINE                      Teach Home Position
     2:  Search Start[1] PR[2]               Start Search Motion. Use Touch Sch 1, PR[2]
     3:J  P[2] 50% FINE                      Teach a search start position
     4:J  P[3] 50% FINE Search[Y]            Search to y direction
     5:J  P[4] 50% FINE                      Teach an another search start position
     6:J  P[5] 50% FINE Search[X]            Search to X direction
     7:  Search End                          Finish Search Motion
     8:J  P[6] 50% FINE                      (If needed) Teach an intermediate point
     9:  Touch Offset PR[2]                  Following points will be offset by PR[2]
    10:L  P[7] 100mm/sec FINE                P[7] is offset by PR[2]
      :  Weld Start[1,1]
    11:L  P[8] 80cm/min FINE                 P[8] is offset by PR[2]
      :  Weld End[1,2]
    12:  Touch Offset End                    Finish offset of positions
```

Fig. 39.5(h) Program example of 2D search (Fillet/Lap search)

Program Example of 2D + Rotation Search

In 2D + Rotation Search, search to two directions are performed, and the search points are two points per one direction. Then taught path is compensated by Touch Offset instruction. Refer to the following program example.

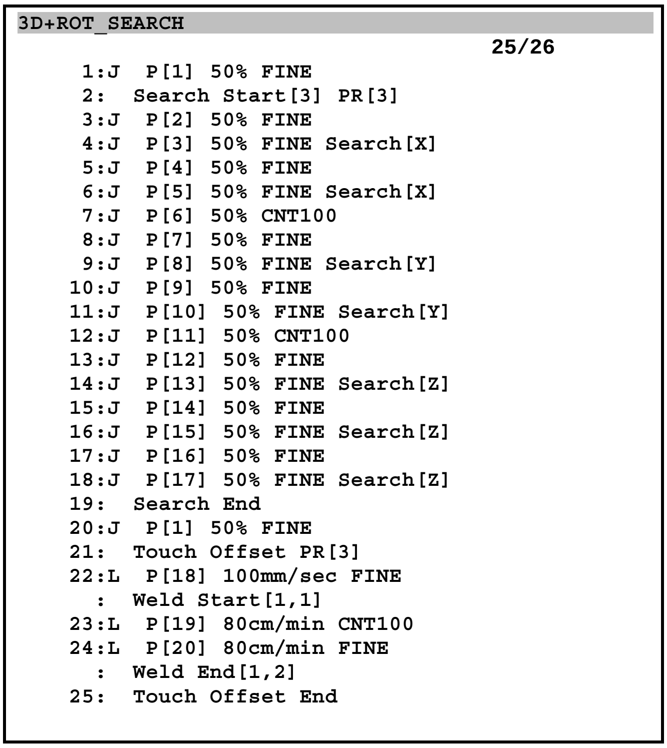


```
2D+ROT_SEARCH
                                    25/26
     1:J  P[1] 50% FINE                      Teach Home Position
     2:  Search Start[3] PR[3]               Start Search Motion. Use Touch Sch 3, PR[3]
     3:J  P[2] 50% FINE                      Teach a search start position
     4:J  P[3] 50% FINE Search[X]            Search to X direction
     5:J  P[4] 50% FINE                      Teach an another search start position
     6:J  P[5] 50% FINE Search[X]            Search to X direction
     7:J  P[6] 50% CNT100                    Teach an intermediate position
     8:J  P[7] 50% FINE                      Teach an another search start position
     9:J  P[8] 50% FINE Search[Y]            Search to Y direction
    10:J  P[9] 50% FINE                      Teach an another search start position
    11:J  P[10] 50% FINE Search[Y]           Search to Y direction
    12:  Search End                          Finish Search Motion
    13:J  P[1] 50% FINE                      Move to Home Position
    14:  Touch Offset PR[3]                  Following points are offset by PR[3]
    15:L  P[11] 100mm/sec FINE               P[11] is offset by PR[3]
      :  Weld Start[1,1]
    16:L  P[12] 80cm/min CNT100              P[12] is offset by PR[3]
    17:L  P[13] 80cm/min FINE                P[13] is offset by PR[3]
      :  Weld End[1,2]
    18:  Touch Offset End                    Finish offset of positions
```

Fig. 39.5(i) Program example of 2D + rotation search (Fillet/Lap search)

Program Example of 3D + Rotation Search

In 3D + Rotation Search, 3 points search to another direction is added to 2D + Rotation Search. Rotations of all directions are also obtained as offset data.



```
3D+ROT_SEARCH
                                              25/26
     1:J  P[1] 50% FINE                                Teach Home Position
     2:  Search Start[3] PR[3]                         Start Search Motion. Use Touch Sch 3, PR[3]
     3:J  P[2] 50% FINE                                Teach a search start position
     4:J  P[3] 50% FINE Search[X]                      Search to X direction
     5:J  P[4] 50% FINE                                Teach an another search start position
     6:J  P[5] 50% FINE Search[X]                      Search to X direction
     7:J  P[6] 50% CNT100                              Teach an intermediate position
     8:J  P[7] 50% FINE                                Teach an another search start position
     9:J  P[8] 50% FINE Search[Y]                      Search to Y direction
    10:J  P[9] 50% FINE                                Teach an another search start position
    11:J  P[10] 50% FINE Search[Y]                     Search to Y direction
    12:J  P[11] 50% CNT100                             Teach an intermediate position
    13:J  P[12] 50% FINE                               Teach an another search start poison
    14:J  P[13] 50% FINE Search[Z]                     Search to Z direction
    15:J  P[14] 50% FINE                               Teach an another search start poison
    16:J  P[15] 50% FINE Search[Z]                     Search to Z direction
    17:J  P[16] 50% FINE                               Teach an another search start poison
    18:J  P[17] 50% FINE Search[Z]                     Search to Z direction
    19:  Search End                                    Finish Search Motion
    20:J  P[1] 50% FINE                                Move to Home Position
    21:  Touch Offset PR[3]                            Following points are offset by PR[3]
    22:L  P[18] 100mm/sec FINE                         P[18] is offset by PR[3]
      :  Weld Start[1,1]
    23:L  P[19] 80cm/min CNT100                        P[19] is offset by PR[3]
    24:L  P[20] 80cm/min FINE                          P[20] is offset by PR[3]
      :  Weld End[1,2]
    25:  Touch Offset End                              Finish offset of positions
```

Fig. 39.5(j) Program example of 3D + rotation search (Fillet/Lap search)

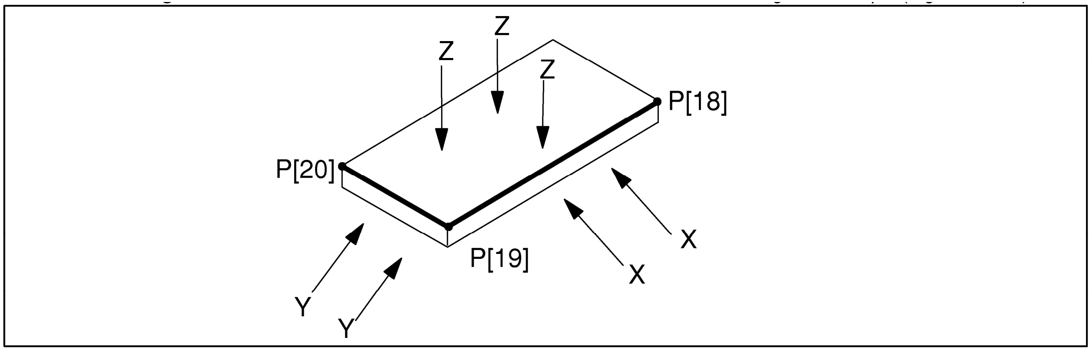


Fig. 39.5(k) Outline of 3D + rotation search

⚠**CAUTION**

For Fillet/Lap, V-Groove, OD/ID search pattern programs, you should perform Search Motion (Search Start – Search End section) first with Master Flag ON to saving information of master positions into TP program. You must establish master positions for all search motion. Then execute Touch Sensing program with Master Flag OFF.

If the program has ever been executed with Master Flag ON, the Touch Sensing program cannot work correctly (Refer to Section 39.6).

39.6 EXECUTION OF TOURCH SENSING PROGRAM

Touch Sensing Mastering

If except simple search pattern is used, offsets generated by touch sensing are relative to the position obtained by mastering. An offset is computed by comparing the position of the current workpiece (offset position) and the original workpiece position (mastering position).

Mastering is necessary once for obtaining the original workpiece position before executing Touch Sensing program. Perform Mastering by referring to Procedure 39-9.

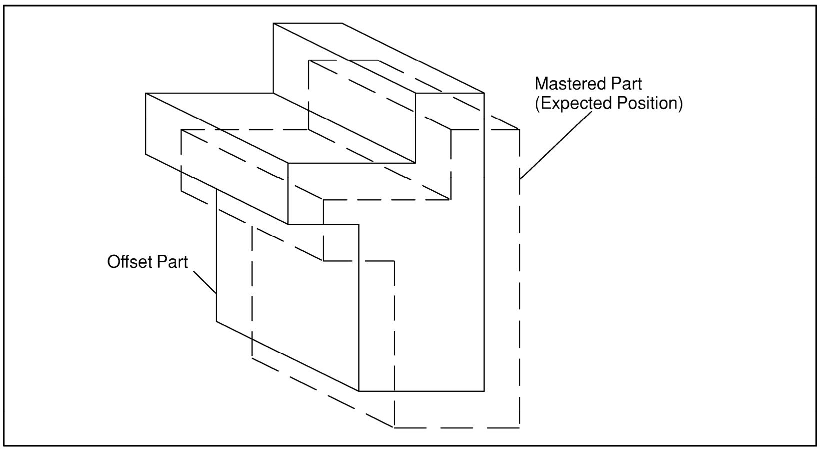


Fig. 39.6(a) Part in mastered position and offset position

> **NOTE**
> A simple search does not require mastering since it produces an absolute position stored in a position register.

Procedure 39-9 Touch Sensing Mastering

Condition

- The workpiece is placed without shift (placed on taught path).
- Touch Sensing Program is already created.
- Except simple search pattern is selected on Touch Schedule.

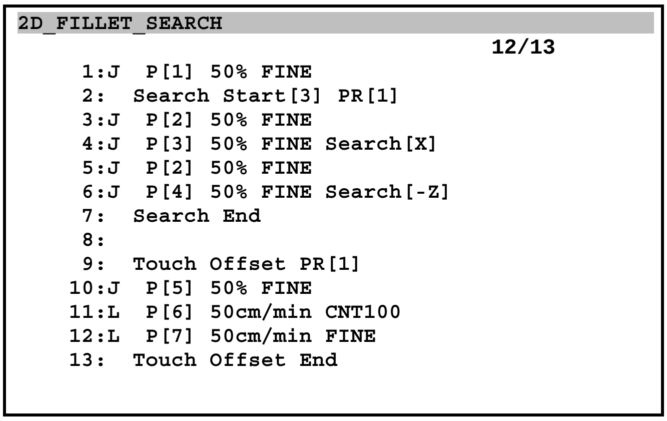
Step

1. Display Touch Schedule Detail screen for used Touch Schedule number by referring to Procedure 39-5.
2. Move the cursor on Master Flag and then press F4, ON. Master Flag becomes ON.
3. Display the program edit for mastering.
4. Execute program between Search Start – Search End instruction (it is also OK with Single Step). Mastering position is stored to the TP program. Then, data in specified position register in Search Start instruction become all zero.
5. Display Touch Schedule Detail screen, move the cursor on Master Flag, and then press F5, OFF. Master Flag becomes OFF.

> **NOTE**
> Incremental search is disabled while the Master Flag is turned ON. If incremental search is not performed, check Master Flag.

Execution of Touch Sensing Program after Mastering

After finishing Mastering, the taught path is compensated every time by executing Search Motion and Touch Offset. The process details of Touch Sensing program after Mastering are explained by referring to the program example of 2D Fillet Search (Fig 39.6(b), (c)).



```
2D_FILLET_SEARCH
                                                   12/13
     1:J  P[1] 50% FINE
     2:  Search Start[3] PR[1]
     3:J  P[2] 50% FINE
     4:J  P[3] 50% FINE Search[X]
     5:J  P[2] 50% FINE
     6:J  P[4] 50% FINE Search[-Z]
     7:  Search End
     8:
     9:  Touch Offset PR[1]
    10:J  P[5] 50% FINE
    11:L  P[6] 50cm/min CNT100
    12:L  P[7] 50cm/min FINE
    13:  Touch Offset End
```

Fig. 39.6(b) Program example of 2D fillet search

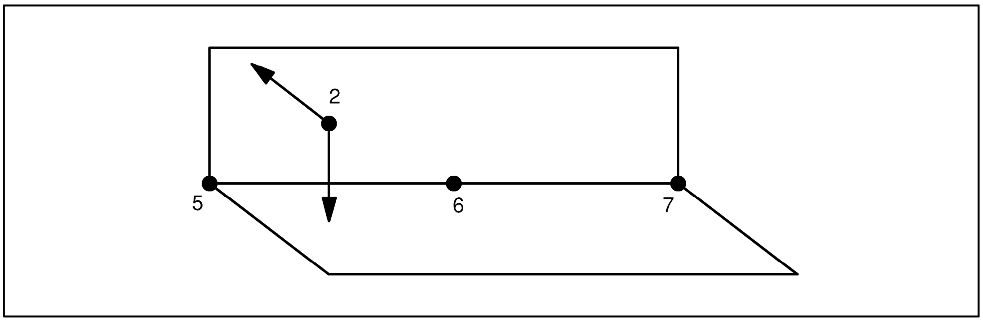


Fig. 39.6(c) Part with one touch start position, 2, and three points along a path, 5, 6, 7

- After finishing Mastering, Mastering position and current position (Offset position) obtained by Search Motion are compared and offset data is calculated when Search Motion (between Search Start – End instruction) is performed with Master Flag OFF. The offset data obtained by this Search Motion becomes relative data between Mastering position and Offset position.

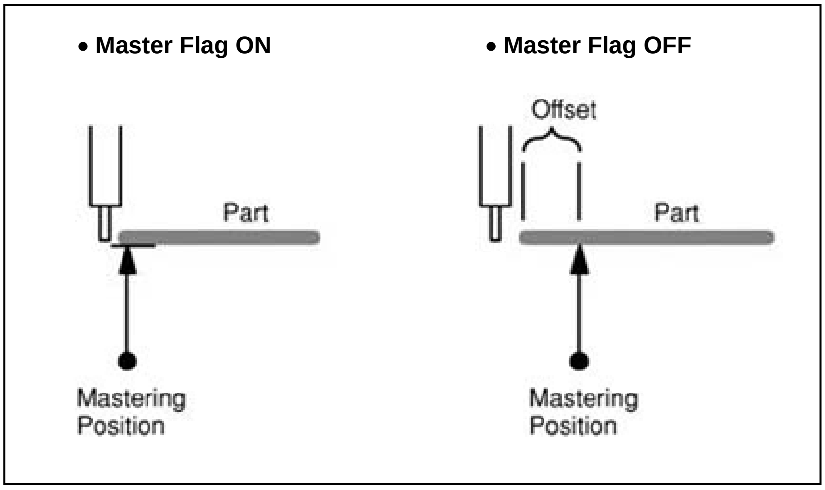


Fig. 39.6(d) Offset value

- After the execution of Search End instruction, offset data is set to position register specified on Search Start instruction.

- After that, compensated motion for current workpiece position is done by applying offset data to all motion instruction placed between Touch Offset Start – End instructions (position 5, 6 and 7 on Program Example).

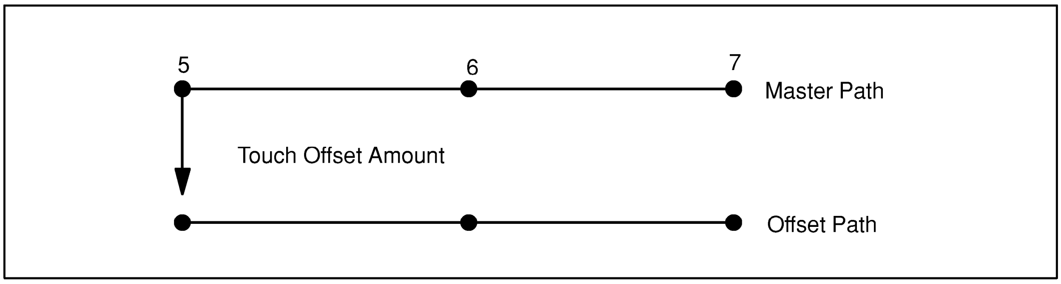


Fig. 39.6(e) Illustration of the path when an offset Is applied

Details of Touch Sensing Program Execution

- If BWD execution is performed for Search Start instruction, Search Motion becomes temporarily disabled. If you want to restart Search Motion, restart the program from the line just before Search Start instruction by FWD execution again.
- Touch Offset finishes when Touch Offset End instruction is executed or program is aborted. On the other hand, Touch Offset continues when program is paused and then resumed.
- If BWD execution is performed for Touch Offset Start instruction, Touch Offset becomes temporarily disabled. If you want to restart Touch Offset, restart the program from the line just before Touch Offset Start instruction by FWD execution again.
- After the FWD execution of Touch Offset Start instruction, if cursor is moved to except between Touch Offset Start – End instructions and then program is restarted, offset specified by Touch Offset Start instruction is applied (that is, the offset is taken over).

Remastering

The touch up procedure described in Section 39.8 should work for most instances where the search start positions do not need to be moved or if the parts do not change drastically.
Remastering is required if the search start positions must be taught again. If the path must be altered significantly, it is recommended to remaster to ensure a correct path.

The procedure of Remastering is not different from normal Mastering (refer to Procedure 39.9). After finishing Remastering, turn Master Flag OFF.

39.7 TOUCHING UP OF TOUCH SENSING PROGRAM

Mastering position data are stored internally into TP program using Touch Sensing. Therefore, take care to change the program. In this section, notifications for touching up new position on Touch Sensing program are explained by referring to Fig. 39.7 (a) and (b).

Touching Up Path Positions

Occasionally the part or its placement on a fixture will change requiring adjustment of the path. The entire process of remastering is not need to accommodate these changes.
When touch up of motion instructions placed between Touch Offset instructions is required, it is necessary to consider the current offset data of workpiece and teach the new master path. Therefore, the operation of touch up must be done after the completion of correct Search Motion and when Touch Offset continues (offset is compensated). In this situation, if touch up is done by jogging the robot, the only difference information between original offset position is reflected to the position data in the program and new correct master path is generated (refer to Fig. 39.7(a), (b)).

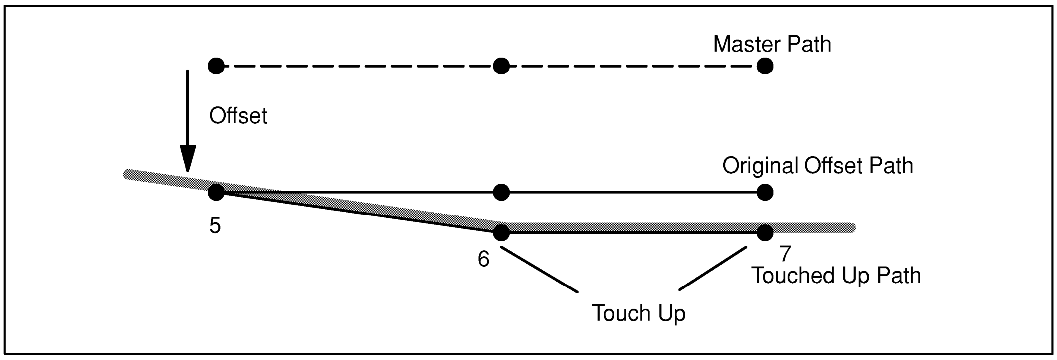


Fig. 39.7 (a) Touch up of position 6 and 7

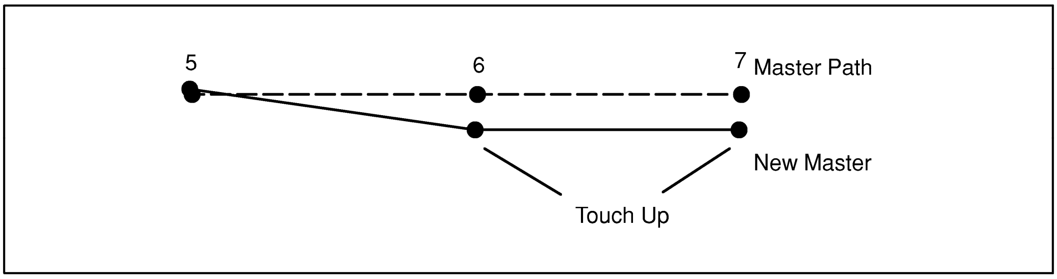
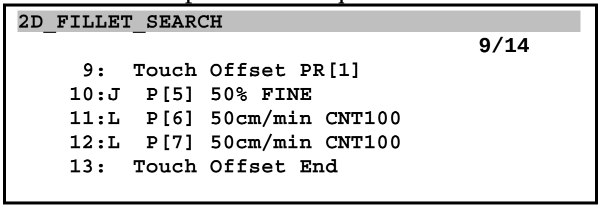


Fig. 39.7 (b) New master path after touch up of position 6 and 7

Procedure 39-10 shows the method to touch up the robot on Touch Sensing program. Additionally, Fig. 39.8(c) shows the example which requires Procedure 39-10 during touch up.



```
2D_FILLET_SEARCH
                                  9/14
   9:  Touch Offset PR[1]
  10:J  P[5] 50% FINE
  11:L  P[6] 50cm/min CNT100
  12:L  P[7] 50cm/min CNT100
  13:  Touch Offset End
```

When you touch up these positions, Procedure 39-10 is required.

Fig. 39.7(c) Example of touch up during touch offset

Procedure 39-10 Touch up of Touch Sensing Program

Condition

- Program edit screen has been displayed.
- Teach pendant is enabled.

Step

1. Execute Search Motion (between Search Start – End instructions) and store the offset data to position register.
2. Execute Touch Offset Start instruction.
3. Forward the program to motion instruction whose position is required to touch up with Single Step.
4. Jog the robot to the position which you want to touch up. Then, move the cursor to position number of current line and press [SHIFT] key and F5, TOUCHUP key. Touch up is done.
5. Touch up all required position between Touch Offset Start – End instructions by Step 4.

⚠**CAUTION**

Do not touch up after BWD execution during execution between Touch Offset Start – End instructions with Single Step. Otherwise, offset data will become wrong value.

Incorrect Touch Up

A common error is to alter the path without the correct offset being applied. Touch up must be done after executing the search and while the Touch Offset is applied.

An example of an incorrect touch up is as follows:

- Execute the program without executing Search Motion.
- Perform touch up the positions between Touch Offset Start – End instructions without executing Touch Offset Start instruction.
- Perform touch up with the position register which has incorrect offset data.

If incorrect touch up is performed, new master position become wrong. Therefore, you cannot obtain the correct compensated path even if Search Motion and Touch Offset are properly done.

Adding New Positions

Additional points can be added in the same manner of touch up Procedure 39-10.

- The search must be completed and an accurate offset must be generated.
- This operation should be done during Touch Offset.
- New position should be added to the offset path.

Fig. 39.8(d) illustrates adding a point to a path.

NOTE

If the program is ABORTED while adding new positions, the offset is cancelled. New positions will be taught as actual locations rather than positions with an offset applied. The results would be similar to what is shown in Fig. 39.8(d) and correct path compensation cannot be obtained.

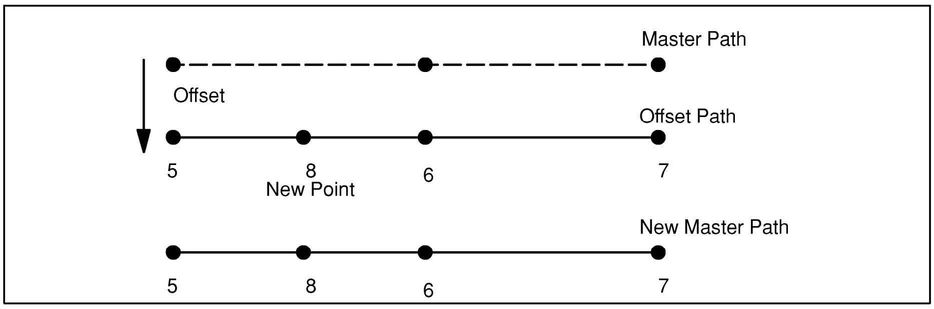


Fig. 39.7(d) Add new point while executing touch offset

Touch Up of Search Start Position

The touch up of a search start position is different from the touch up of the Touch Offset path position. If the search start position is touched up, remastering must be executed. But, if the search start position is moved along the axis of the search direction, remastering is not necessary.

Example that does not require Remastering

If the search start position is too close to the part due to poor programming, changes in the part location, or a change in the part location, then the search start position needs only to be moved back along the search direction. This can be accomplished with no effect on the path positions and remastering will not be required.

The following example shown in Fig. 39.7(e) shows a part and search start position.

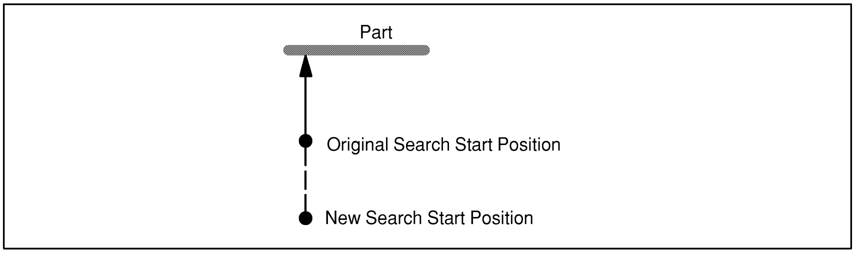


Fig. 39.7(e) Moving a search start position along the search direction

Example which requires Remastering

The example shown in Fig. 39.7(f) shows the search start position moved to the position separated the axis of the search direction. If the search position is moved off the axis of the search direction, Remastering is required. About Remastering, refer to Section 39.7.

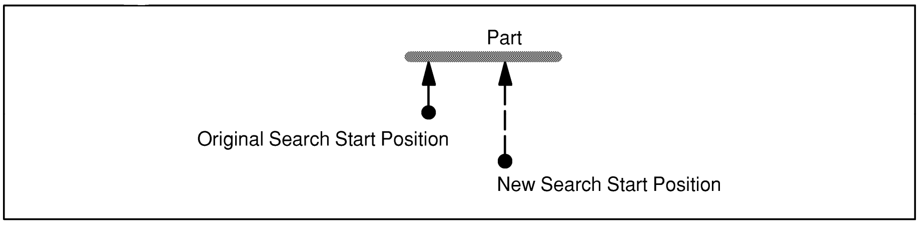


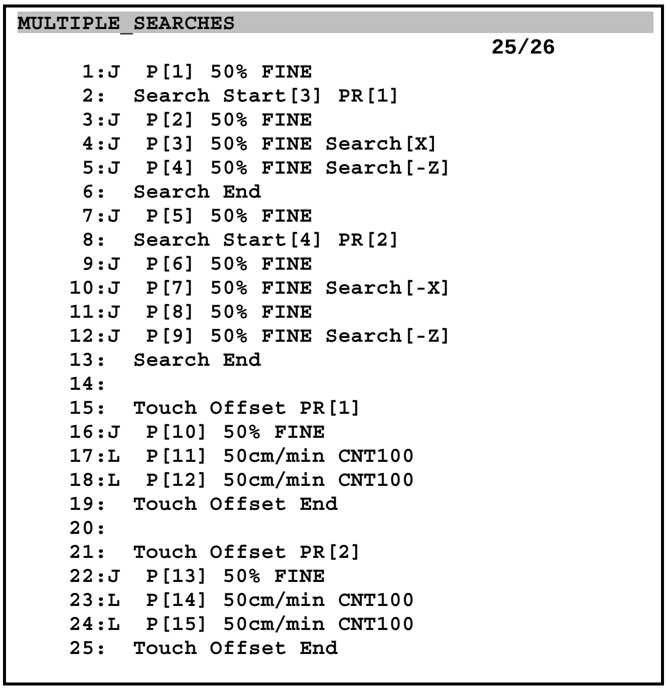
Fig. 39.7(f) Search start position Is separated from original search direction axis

39.8 MULTIPLE SEARCHES

Complex program can have multiple searches generating several offsets.

The following program example shown in Fig. 39.9(a) shows two searches that can be performed for complex shapes. When search is performed for that complex workpiece, Search Motion may not be able to complete by one Touch Frame. Therefore, multiple searches are required.

- The first search stores the offset data in position register 1 with positions 10, 11, 12 using the offset.
- The second search stores offset data in position register 2 with positions 13, 14, 15 using the offset.



```
MULTIPLE_SEARCHES
                                              25/26
     1:J  P[1] 50% FINE
     2:  Search Start[3] PR[1]
     3:J  P[2] 50% FINE
     4:J  P[3] 50% FINE Search[X]
     5:J  P[4] 50% FINE Search[-Z]
     6:  Search End
     7:J  P[5] 50% FINE
     8:  Search Start[4] PR[2]
     9:J  P[6] 50% FINE
    10:J  P[7] 50% FINE Search[-X]
    11:J  P[8] 50% FINE
    12:J  P[9] 50% FINE Search[-Z]
    13:  Search End
    14:
    15:  Touch Offset PR[1]
    16:J  P[10] 50% FINE
    17:L  P[11] 50cm/min CNT100
    18:L  P[12] 50cm/min CNT100
    19:  Touch Offset End
    20:
    21:  Touch Offset PR[2]
    22:J  P[13] 50% FINE
    23:L  P[14] 50cm/min CNT100
    24:L  P[15] 50cm/min CNT100
    25:  Touch Offset End
```

Fig. 39.8(a) Program example of multiple searches

If a position of the taught path is to be touched up, the corresponding search must be performed.

- The first search must be executed to obtain an accurate offset.
- The offset is applied and the positions, 10, 11, and 12 can be touched up as normal.
- The master will be correctly updated.

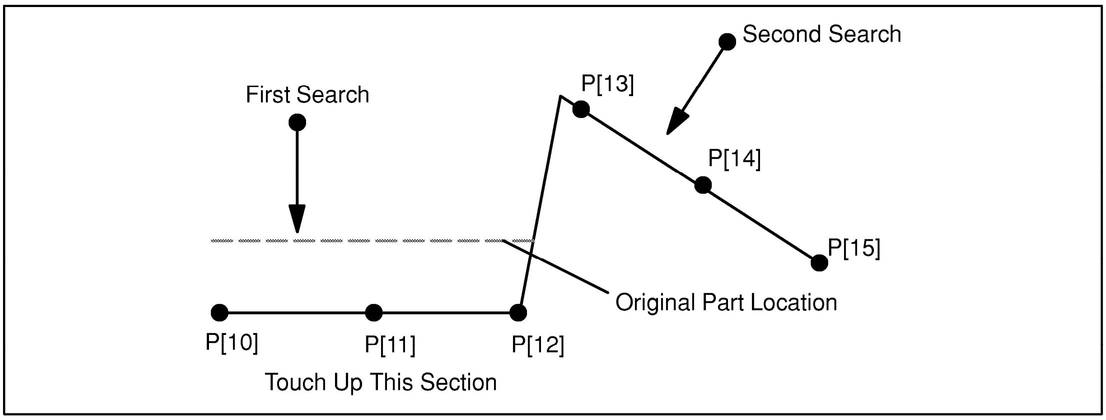


Fig. 39.8(b) Touch up on multiple searches

39.9 COORDINATED TOUCH SENSING

Touch Offset by Touch Sensing can also apply to a motion sentence which has a coordinated motion instruction. To achieve this, you must set Touch Frame and Touch Schedule for Coordinated Touch Sensing. The setup and program are changed by search pattern (Simple Search, or others).

Coordinated Touch Sensing except Simple Search

Fig. 39.10(a) shows Search Motion of leader group (Group 2: Positioner) and follower group (Group 1: Robot) for 2D+Rotation with coordinated motion. Search Motion is done based on coordinated frame of leader group. Touch Offset motion after Search Motion is shown in Fig. 39.9(b).

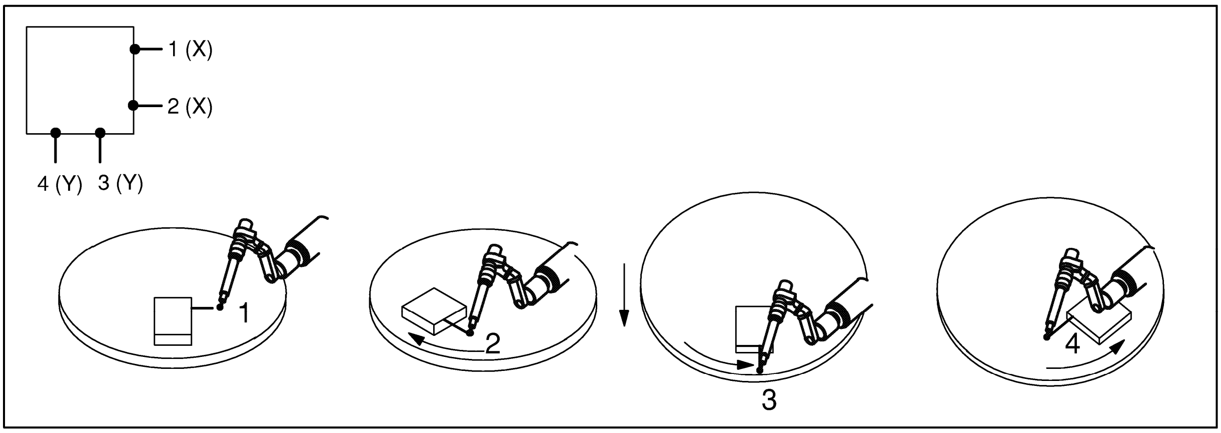


Fig. 39.9(a) 2D+rotation search with coordinated motion

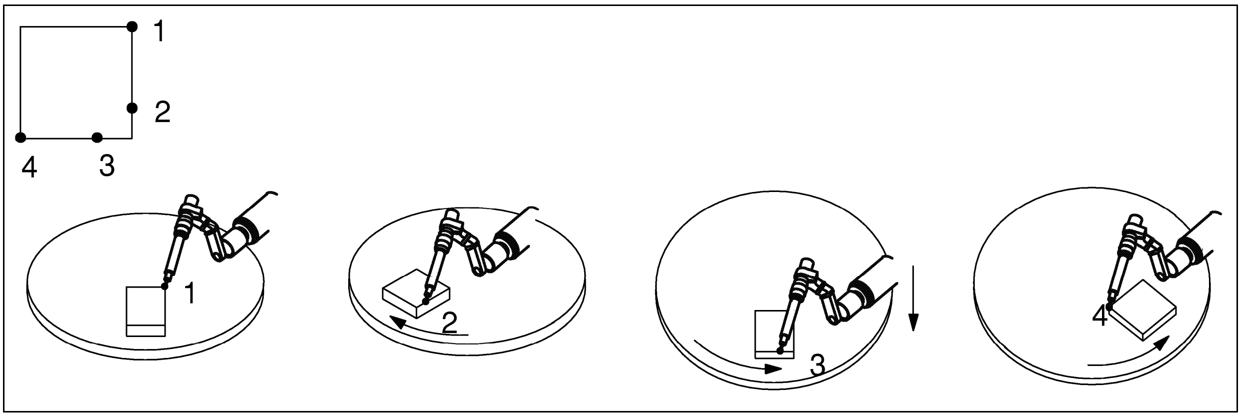
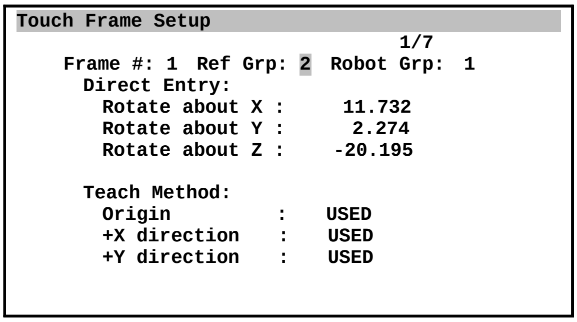


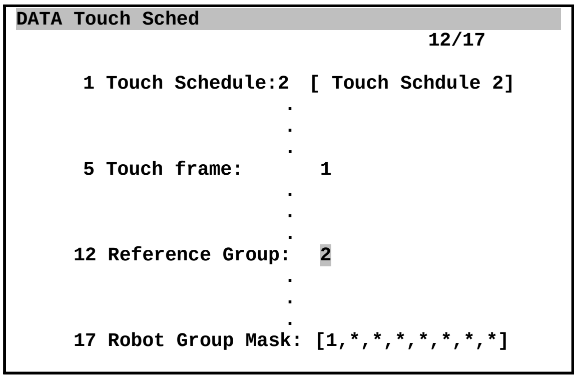
Fig. 39.9(b) Touch offset with coordinated motion

First, Touch Frame setup is shown.
Enter the group number of robot group (follower group) that performs Search Motion and Touch Offset to “RoBot Grp” item. After that, enter the group number of leader group of coordinated motion to “Ref Grp” item. That is, if Coordinated Touch Sensing is performed, “Ref Grp” number and “Robot Grp” number” in Touch Frame are different. About more detail, refer to Procedure 39-3 and 39-4.
Before the setup, you should already complete the calibration of coordinated pair between leader group and follower group in coordinated motion. If you change “Ref Grp” item on Touch Frame without calibration of coordinated pair, “Referenced group does not exist” message will be displayed. About the calibration of coordinated pair, please refer to Coordinated Motion Function OPERATOR’S MANUAL (B-83484EN).

Additionally, “Ref Grp” number and “Robot Grp” number on Touch Frame must be also entered on Touch Schedule. Set the robot group for Touch Sensing to “Robot Group Mask” item in Touch Schedule Detail screen. Next, enter Touch Frame number to “Touch Frame” item. Then, enter the same number as “Ref Grp” number (Leader group number) on Touch Frame to “Reference Group” item.



```
Touch Frame Setup
                                         1/7
    Frame #: 1  Ref Grp: 2  Robot Grp:  1
      Direct Entry:
        Rotate about X :     11.732
        Rotate about Y :      2.274
        Rotate about Z :    -20.195

      Teach Method:
        Origin         :   USED
        +X direction   :   USED
        +Y direction   :   USED
```

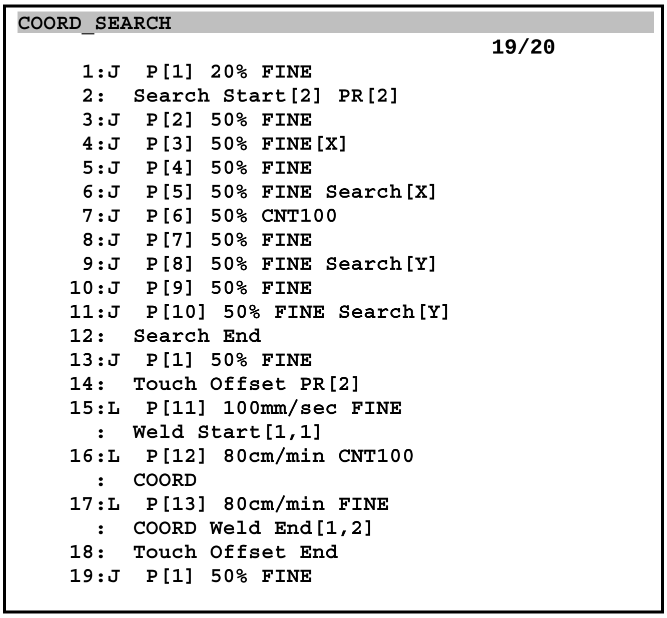


```
DATA Touch Sched
                                         12/17

      1 Touch Schedule:2  [ Touch Schdule 2]
                       .
                       .
                       .
      5 Touch frame:      1
                       .
                       .
                       .
     12 Reference Group:   2
                       .
                       .
                       .
     17 Robot Group Mask: [1,*,*,*,*,*,*,*]
```

Fig. 39.9(c) Setup example of touch frame and touch schedule (Coord, except simple search)

Fig. 39.9(d) is the program example. Search direction is derived from the direction of leader group (positioner in the figure). Even if leader group moves, search direction is not changed relatively. It is also possible to teach the motion instruction which is not related to Search Motion (refer to Line 7 of Fig. 39.9(d)). The offset data obtained by Search Motion is compensated by the direction of leader group.

Group Mask in the Program[1,1,*,*,*,*,*,*]



```
COORD_SEARCH
                                    19/20
     1:J  P[1] 20% FINE
     2:  Search Start[2] PR[2]
     3:J  P[2] 50% FINE
     4:J  P[3] 50% FINE[X]
     5:J  P[4] 50% FINE
     6:J  P[5] 50% FINE Search[X]
     7:J  P[6] 50% CNT100
     8:J  P[7] 50% FINE
     9:J  P[8] 50% FINE Search[Y]
    10:J  P[9] 50% FINE
    11:J  P[10] 50% FINE Search[Y]
    12:  Search End
    13:J  P[1] 50% FINE
    14:  Touch Offset PR[2]
    15:L  P[11] 100mm/sec FINE
      :  Weld Start[1,1]
    16:L  P[12] 80cm/min CNT100
      :  COORD
    17:L  P[13] 80cm/min FINE
      :  COORD Weld End[1,2]
    18:  Touch Offset End
    19:J  P[1] 50% FINE
```

Teach Home Position (Leader/Follower groups)
Start Search Motion: Touch Sch 2, PR[2]
Teach a search start position(Leader/Follower)
Search to X direction (Only follower group)
Teach an another search start point(Leader/Follower)
Search to X direction (Only follower group)
Teach an intermediate point (Leader/Follower)
Teach an another search start point(Leader/Follower
Search to Y direction (Only follower group)
Teach an another search start point(Leader/Follower
Search to Y direction (Only follower group)
Finish Search Motion
Move to Home Position
Following points are offset by PR[2]
P[11] is offset by PR[2]

P[12] is offset by PR[2]

P[13] is offset by PR[2]

Finish offset of positions
Teach Home Position

Fig. 39.9(d) 2D + rotation search with coordinated motion

NOTE

When reference group setup is performed for coordinated motion, mastering information of all groups related to reference group are stored in TP program. If you change reference group on Touch Frame or Touch Schedule, please perform remastering.

Coordinated Touch Sensing with Simple Search

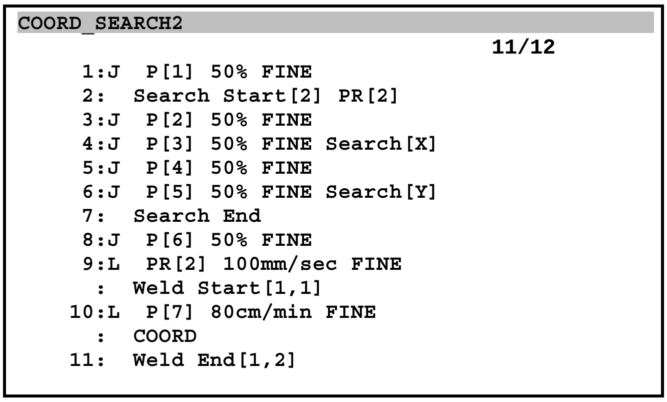
Touch Frame for Simple Search can select world frame of follower group or coordinated frame of leader group as basis.

If you select world frame of follower group as basis, enter the group number of follower group. If you select coordinated frame of leader group as basis, enter the group number of leader group. In this case, you must complete the calibration of coordinated pair. Additionally, absolute position data in position register by Search Motion is based on follower group.

In Touch Schedule for Simple Search, reference group number must be the same number as Robot Group Mask.

Following Fig. 39.9(e) is the program example of Simple Search with coordinated motion.

Group Mask of TP Program[1,1,*,*,*,*,*,*]



```
COORD_SEARCH2
                                            11/12
     1:J  P[1] 50% FINE
     2:  Search Start[2] PR[2]
     3:J  P[2] 50% FINE
     4:J  P[3] 50% FINE Search[X]
     5:J  P[4] 50% FINE
     6:J  P[5] 50% FINE Search[Y]
     7:  Search End
     8:J  P[6] 50% FINE
     9:L  PR[2] 100mm/sec FINE
      :  Weld Start[1,1]
    10:L  P[7] 80cm/min FINE
      :  COORD
    11:  Weld End[1,2]
```

Teach Home Position (Leader/Follower groups)
Start Search Motion, Touch Sch 2, PR[2]
Teach a search start position (Leader/Follower)
Search to X direction (Only follower group)
Teach an another search start point(Leader/Follower)
Search to Y direction (Only follower group)
Finish Search Motion
Teach an intermediate point (Leader/Follower groups)
Move the robot to PR[2]
Start Arc Welding
Coordinated motion

Fig. 39.9(e) Program example of simple search with coordinated motion

⚠CAUTION

Search Motions and Touch Offsets for coordinated motion cannot be mixed with non-coordinated motion. That is, you cannot do searches for coordinated motion, then do searches for non-coordinated motion, and then use the offset from the coordinated motion search. You must use the offsets for coordinated motion searches before doing any non-coordinated searches. Similarly, you must use any offsets from non-coordinated motion searches before doing any coordinated motion searches.

NOTE

The leader group is not allowed to move between the searches.

40 TOUCH SKIP FUNCTION

Touch skip function can watch the disturbance of any plural axes at the same time and when any axes' disturbance exceed threshold high-speed skip can be executed.
The threshold to judge the touch can be set in the special screen for touch skip function.

When touch skip function is ordered,

- "SETUP/ TOUCH SKIP" screen is able to be shown.
- Touch skip programs are able to be called.

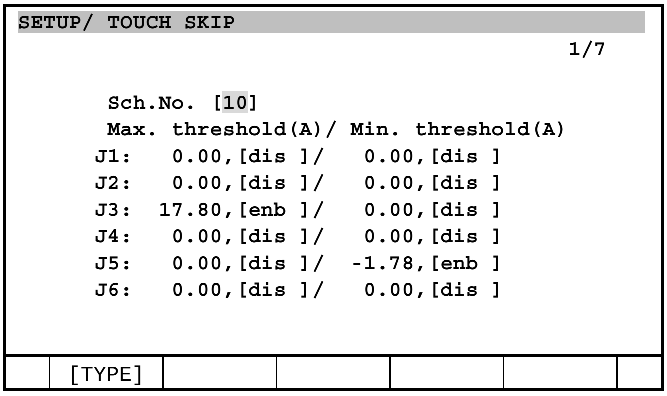
Note that system variable $MISC_MSTR.$HPD_ENB must be TRUE when Touch skip is used. (This variable is FALSE by default.)
If $MISC_MSTR.$HPD_ENB is not TRUE, please set it to TRUE by system variable screen or parameter instruction, and cycle power.

To use this function, Touch skip function option (A05B-2600-J921) is required.
Some robot models do not support this function.

40.1 TOUCH SKIP SCREEN

In touch skip screen, the conditions to judge the touch can be set.

- Show Touch skip screen by the operation of [MENU] key → "SETUP" → F1 key → "Touch Skip".



```
SETUP/ TOUCH SKIP
                                                1/7

      Sch.No. [10]
      Max. threshold(A)/ Min. threshold(A)
     J1:    0.00,[dis ]/    0.00,[dis ]
     J2:    0.00,[dis ]/    0.00,[dis ]
     J3:   17.80,[enb ]/    0.00,[dis ]
     J4:    0.00,[dis ]/    0.00,[dis ]
     J5:    0.00,[dis ]/   -1.78,[enb ]
     J6:    0.00,[dis ]/    0.00,[dis ]
```

- To change the schedule number, enter a value in "Sch. No." field.
- In lines below, set values and enable/disable of thresholds for disturbance of each axis.
  Check the disturbance during the motion by STATE → AXES → DISTURBANCE screen and set thresholds that is added margins if necessary.

In the example above, a condition of "J3 disturbance is larger than 17.8[A] or J5 disturbance is smaller than –1.78[A]" is set.

40.2 TOUCH SKIP PROGRAM

Following programs are able to be called.

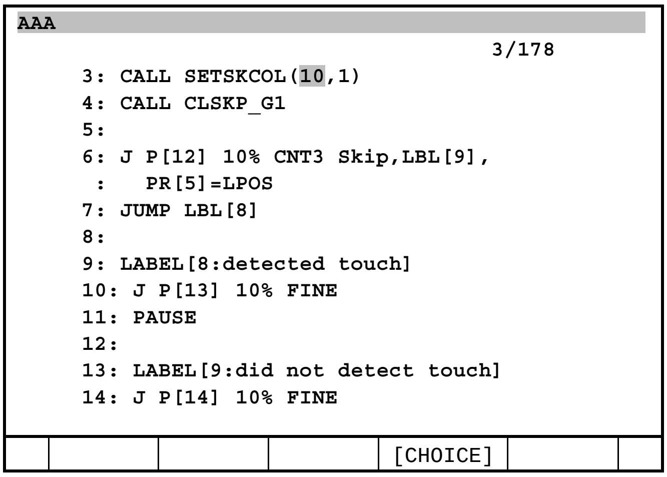
- SETSKCOL(s, g)

This is a program to set schedule No. [s], which is set on touch skip screen, to threshold of group [g].

- CLSKP_Gg

These are programs to set skip condition as "disturbance exceeds threshold in an axis of group [g]".

For example, you can use these programs as follows.



```
AAA
                                        3/178
     3: CALL SETSKCOL(10,1)
     4: CALL CLSKP_G1
     5:
     6: J P[12] 10% CNT3 Skip,LBL[9],
      :   PR[5]=LPOS
     7: JUMP LBL[8]
     8:
     9: LABEL[8:detected touch]
     10: J P[13] 10% FINE
     11: PAUSE
     12:
     13: LABEL[9:did not detect touch]
     14: J P[14] 10% FINE
```

- In line3, schedule No. [10] which is set on touch skip screen is set on threshold of group [1].
- In line4, skip condition "disturbance exceeds threshold in an axis of group [1]" is set.
  Same as SKIP CONDITION instruction, this condition has been effective until program ends or another condition is set.
- A motion with high-speed skip instruction in line6 works all the same as high-speed skip instruction of standard function.
  When disturbance exceeds threshold in an axis of group [1], the robot stops and then the next line (line7) is executed.
  When disturbance does not exceed threshold in any axis of group [1] and the robot reaches destination point, the jump to the label [9] is executed.

40.3 CAUTIONS

- To decrease impact of stopping when skip condition is satisfied, speed limit is set for the motion with touch 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)

- Some robot models decreased stop when high-speed skip instruction is executed. With such a robot, touch skip function can not be used. When touch skip function is executed with such a robot, the following alarm is displayed.

  MOTN-561 Touch skip NOT available (G:group number)

- Touch skip can not be used with skip instruction (Skip, LBL). When touch skip is executed with skip instruction (Skip, LBL), the following alarm is displayed. When you use touch skip function, you must teach high-speed skip instruction (Skip, LBL, PR).

  MOTN-561 Touch skip NOT available (G:group number)

41 MROT INSTRUCTION

MROT (Minimal Rotation) is a motion option to be used with the Joint motion type, or Cartesian motion with the WJNT option. It generates the shortest joint angle move for the wrist axes within axes limits during Joint and WJNT motion, and ignores the turn numbers of the wrist. For example, the start and destination angles are 150° and -130°, respectively. In the case, the shortest move is the motion from 150° to 230°. Because, 230° and -130° are the same position for the axis physically.

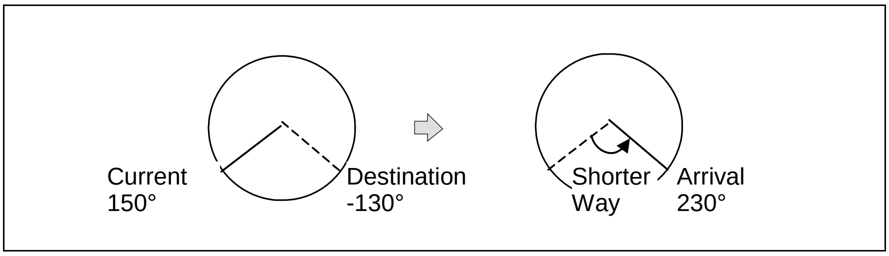


Fig. 41 MROT instruction

To use this function, MROT Instruction optionis required.

41.1 HOW TO USE MROT

Add MROT instruction to Joint or WJNT motion.

J P[1] 100% FINE MROT
L P[2] 2000mm/sec CNT100 WJNT MROT

41.2 LIMITATIONS

The MROT motion option has following limitations.

- This function is supported only in Handling Tool.
- Before MROT can take effect, the destination position of the Joint or WJNT motion must be represented in Cartesian space. Otherwise, if the destination position is represented in Joint angles, the system will attempt to reach the specified destination joint angles regardless of the MROT option.
- MROT generates shortest joint move within axes limits. For example, the upper limit of wrist axis is 180°, and the start and destination angle are 150° and -130°, respectively. In this case, a warning massage "MOTN-330 MROT Limit Warn" will be posted and the axis will move from 150° to -130°. Because 230° is out of limit.

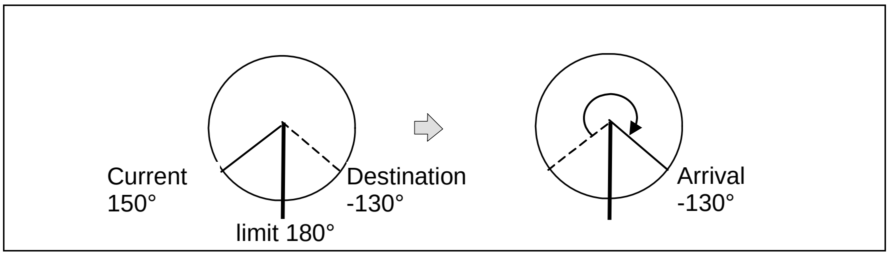


Fig. 41.2 Shorter way is out of limit.

- Single step Backward motion may not be the same as forward motion.

For example:

```
1: J P[1] 100% FINE
2: J P[2] 100% FINE MROT
3: J P[3] 100% FINE
```

If the actual reached joint angles at P[2] are different from the taught joint angles (in terms of turn number), then backward motion from P[3] to P[2] would be different from the forward motion from P[2] to P[3]. Because the backward motion from P[3] to P[2] is not MROT motion, the arrival position is taught position of P[2].

42 ROBOT ISOLATION FUNCTION

Robot Isolation function can make certain robot be isolated state by the flick of a switch when multiple robot is controlled by single controller. In teaching in the system with multiple robot, when just partial robots are taught, it is possible to keep the other robots unmoving. On this account, the user can have safety and convenience in teaching.

This function is usable if the hardware of the switch for Robot Isolation is attached. No software option is required.

42.1.1 Specification

- Isolated state
  Robot turns into machine lock state by isolating. Servo power is turned off and robot comes into unmoving state. And the user can't teach new point to isolated robot. By unlocking isolated state, robot turns into normal state.

- Method of operating
  Isolation switch is prepared up to 4 on the operation panel. And by Isolation switch, the user can change certain robot state between isolated and connected. When robot is isolated, robot turns into machine lock state after alarm is displayed. And when robot is connected, robot turns into normal state after alarm is displayed.

NOTE

1 If robot state is changed from isolated to connected in power-off, robot remains in machine lock state after power is turned on. In this instance, it is needed to change robot state by the Teach Pendant to be normal state.

2 If robot state is changed from connected to isolated in power-off, robot turns into machine lock state during power up after power is turned on.

- Setting
  Each Isolation switch corresponds certain motion group. User can see correspondence between Isolation switch and motion group by system variable $ROBOT_ISOLC.

42.2 METHOD OF OPERATING

42.2.1 Operation Panel

As is often the case, Isolation switch is mounted on Operation panel.

By turning switch ENABLE side, robot become operable.
By turning switch DISABLE side, robot become isolated.

42.2.2 DCS Safety Signal: RPI

If DCS Safe I/O connect function (J568) is ordered, DCS safety signal: RPI is usable.

It is possible to see the state of Isolation switch by RPI[1-4].

RPI is ON: Isolation switch is ENABLE side (connected).

RPI is OFF: Isolation switch is DISABLE side (isolated).

Isolation switch 1 - RPI[1]
Isolation switch 2 - RPI[2]
Isolation switch 3 - RPI[3]
Isolation switch 4 - RPI[4]

42.3 SYSTEM VARIABLE: $ROBOT_ISOLC

Depending on ON/OFF of each Isolation switch, corresponding motion group state become normal / machine lock. Correspondence between Isolation switch and motion group is decided by system variable $ROBOT_ISOLC.
Which robot's power current to servo is turned ON/OFF by Isolation switch is decided by hardware configuration. Therefore, it is decided uniquely which motion group should be machine lock state by software when certain isolation switch is turned.
So $ROBOT_ISOLC is needed to be set just one time when robot system is established.

$ROBOT_ISOLC have 4 alignment, and each one corresponds one Isolation switch. If the number of Isolation switch is smaller than 4, $ROBOT_ISOLC alignments are available as many as the number of Isolation switch, and the other alignments is ineffective.

SW1: $ROBOT_ISOLC[1]
SW2: $ROBOT_ISOLC[2]
SW3: $ROBOT_ISOLC[3]
SW4: $ROBOT_ISOLC[4]

The each bit of $ROBOT_ISOLC value is equivalent to each motion group.

Bit31	・・・	Bit1	Bit0
Motion Group32	・・・	Motion Group2	Motion Group1

For example, the default values of $ROBOT_ISOLC are set as the following, it is assumed that one Isolation switch is equivalent to one motion group.

Example1 (Default)

SW1: $ROBOT_ISOLC[1] = 1
SW2: $ROBOT_ISOLC[2] = 2
SW3: $ROBOT_ISOLC[3] = 4
SW4: $ROBOT_ISOLC[4] = 8

In case that SW1 is equivalent to motion group 1, 3,
and SW2 is equivalent to motion group 2, 4,
$ROBOT_ISOLC value is the following.

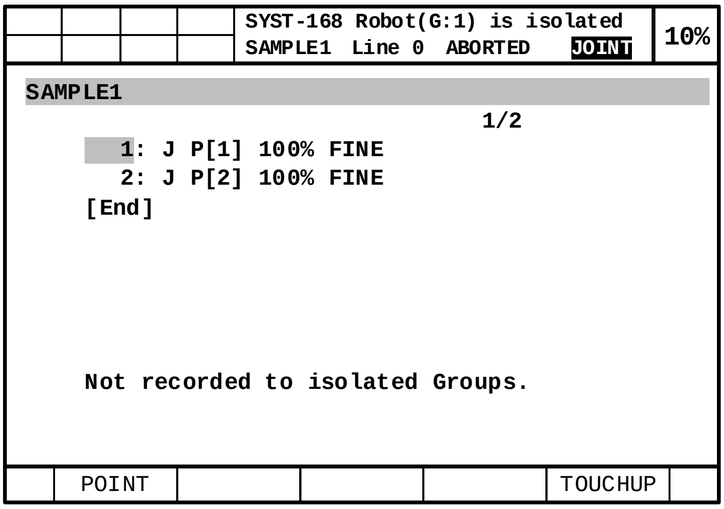
Example 2

SW1: $ROBOT_ISOLC[1] = 5
SW2: $ROBOT_ISOLC[2] = 10
SW3: $ROBOT_ISOLC[3] = 0
SW4: $ROBOT_ISOLC[4] = 0

42.3.1 Teaching to Motion Instruction

When the user teaches motion instruction, current position isn’t recorded for isolated robot.
And when teaches target position again by [SHIFT] key + F5 “TOUCHUP”, position isn’t recorded similarly.

Then, the message that informs isolated group can’t be taught is displayed.



```
SYST-168 Robot(G:1) is isolated
SAMPLE1  Line 0  ABORTED    JOINT      10%

SAMPLE1
                                   1/2
     1: J P[1] 100% FINE
     2: J P[2] 100% FINE
   [End]



   Not recorded to isolated Groups.
```

43 ANTI-DEFLECTION FOR EXTERNAL FORCE

Anti-Deflection for External force suppresses the replacement of TCP due to external force for application which works under external force. For example, this function is usable for tool which drills a hole by pressing the work with a cylinder and fixing the robot to the work in advance. This function compensates the replacement due to reaction force in which presses the work with a cylinder, and improves accuracy of drilling. To use this function, Anti-Deflection for External force function is required.

NOTE

Only M-900*i*B/700 is supported.

When robot is acted by external force, TCP is displaced by RV reducer torsion and so on. This function estimates displacement of TCP and compensates it.

NOTE

No special hardware is required.

43.1 PREPARATION TO CREATE COMPENSATION PROGRAM

Two KAREL programs, GETBQCUR and CALTANG, are loaded if this function is ordered. Compensation amount is calculated by executing GETBQCUR before being acted by external force and executing CALTANG after being acted by external force. Please enter following arguments to these KAREL programs.

GETBQCUR (n)

n: Group number of target robot.

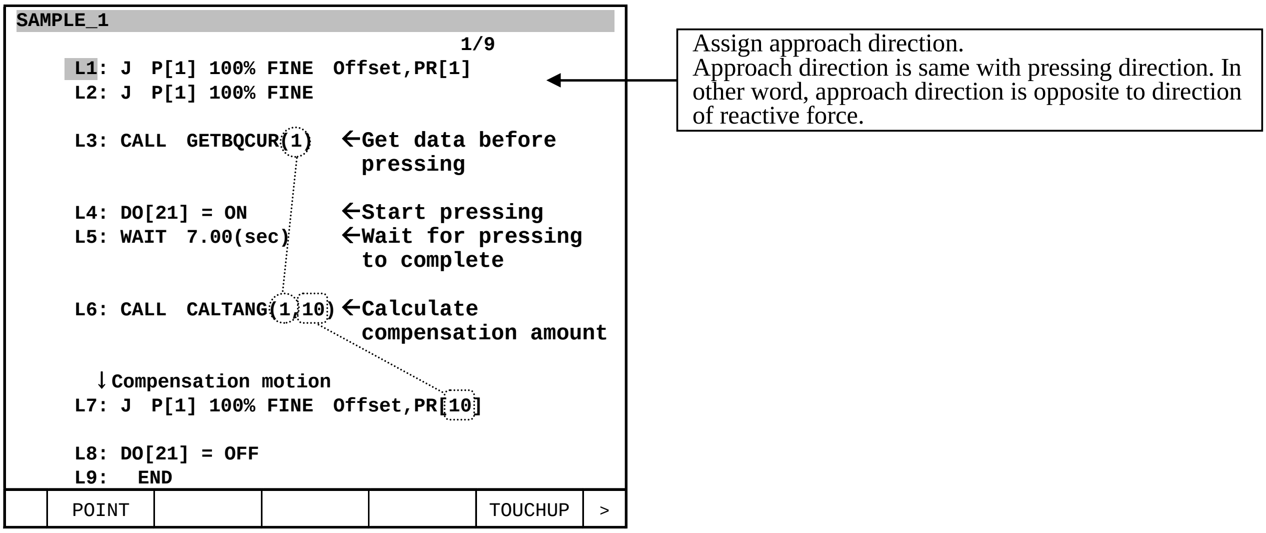
CALTANG (n, m)

n: Group number of target robot.
Set same number with group number assigned at GETBQCUR.

m: Position register number for output compensation amount

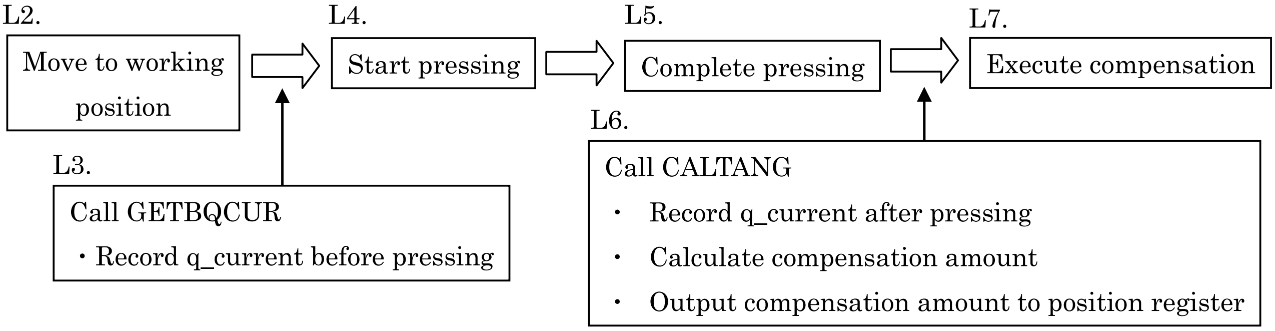
43.2 HOW TO COMPENSATE

This section shows sample program and flow.



```
SAMPLE_1
                                        1/9
    L1: J  P[1] 100% FINE  Offset,PR[1]
    L2: J  P[1] 100% FINE

    L3: CALL  GETBQCUR(1)   ←Get data before
                              pressing

    L4: DO[21] = ON         ←Start pressing
    L5: WAIT  7.00(sec)     ←Wait for pressing
                              to complete

    L6: CALL  CALTANG(1,10) ←Calculate
                              compensation amount

      ↓Compensation motion
    L7: J  P[1] 100% FINE  Offset,PR[10]

    L8: DO[21] = OFF
    L9:   END
   POINT                    TOUCHUP  >
```

Assign approach direction.
Approach direction is same with pressing direction. In other word, approach direction is opposite to direction of reactive force.


L2.
Move to working position
L4.
Start pressing
L5.
Complete pressing
L7.
Execute compensation
L3.
Call GETBQCUR
・Record q_current before pressing
L6.
Call CALTANG
・ Record q_current after pressing
・ Calculate compensation amount
・ Output compensation amount to position register

Following explanation is details for each line.

L1: Specify approach direction to approach from constant direction. Approach direction is same with pressing direction. In other word, approach direction is opposite to direction of reactive force.
L2: Move to working position.
L3: Call KAREL program GETBQCUR and get data before pressing.
L4: Start pressing.
L5: Wait for the completion of pressing.
L6: Call KAREL program CALTANG and get data after pressing and calculate compensation amount of each axis. Compensation amount is output to assigned position register with JOINTPOS type.
L7: Compensate using direct offset condition instruction. Please assign position register number assigned at argument2 of CALTANG.
-----Task to the application -----
L8: Finish pressing
L9: Finish compensation sequence

⚠**CAUTION**
Data type of compensation amount output at position register is JOINTPOS.

43.3 SETUP CUSTOMIZATION

It is possible to change following variables from KAREL variable screen. The method to display KAREL variable is following.

Step

1 Press [MENU] key.
2 Press [0 NEXT] and select [6 SYSTEM].
3 Press F1 [TYPE], and select [SYSTEM Variable] from the menu.
4 Place the cursor on $KAREL_ENB, then key in “1” and press [ENTER] key.
5 Select CALTANG of KAREL program in SELECT screen.
6 Press [DATA] key.
7 Press F1, [TYPE] and select “KAREL Vars”.

Please refer to Subsection 9.20.6 of B-83284EN R-30*i*B/R-30*i*B Mate Controller Operator’s manual (Basic operation) for details of KAREL variable screen.

Name	Type	Explanation

offset_mode	INTEGER	Add offset to compensation amount with following type.
		offset_mode = 1 :XYZWPR
		offset_mode = 2 :JOINT (Default : 0)

offset	ARRAY[MAX_GRP_AXES] OR REAL	Offset amount. If XYZWPR type is selected (offset_mode=1), then offset[1] to offset[6] correspond to such as X:offset[1], Y:offset[2], … , P:offset[5],R:offset[6]. If JOINTPOS type is selected (offset_mode=2), then offset[1] to offset[6] correspond to such as offset[axis number]. Offset amount is calculated based on assigned frame. (Default : 0)
utool_num	INTEGER	Tool frame number. (Default : Selected number)
ufram_num	INTEGER	User frame number. (Default : Selected number)
max_comp_deg	REAL	high limit of compensation amount (Default : 3.0)
spr_const	ARRAY[6] OF REAL	Spring constant of each axis. There is no need to change this variables unless the circumstances are exceptional. (Default : Tuned value)
backlush	ARRAY[6] OF REAL	Backlush of each axis. There is no need to change this variables unless the circumstances are exceptional. (Default : Tuned value)
margin	REAL	Final compensation amount of each axis is given by multiplying compensation amount by margin. (default : 1.0)

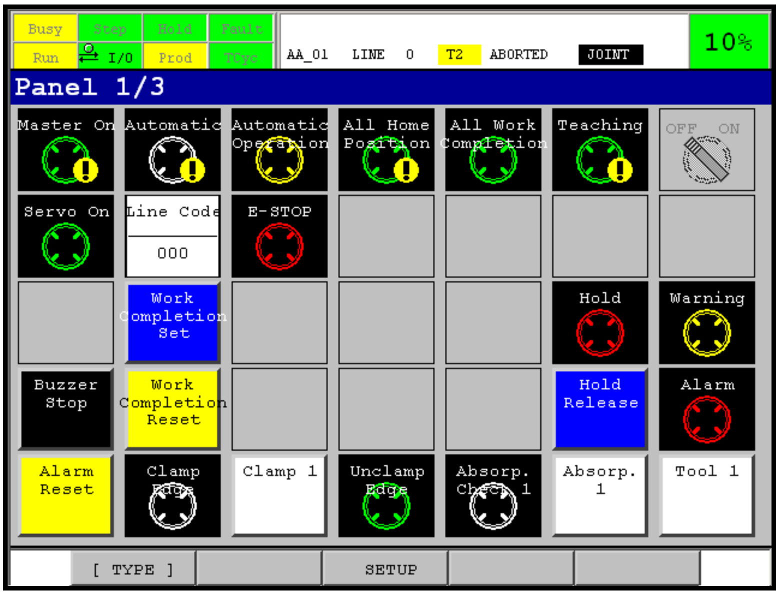
⚠CAUTION
When restarting controller, listed above KAREL variables return to uninitialized variables. These variables return to default when KAREL program CALTANG is executed. If you want to use variables before restarting, please reconfigure it.

43.4 RESTRICTIONS

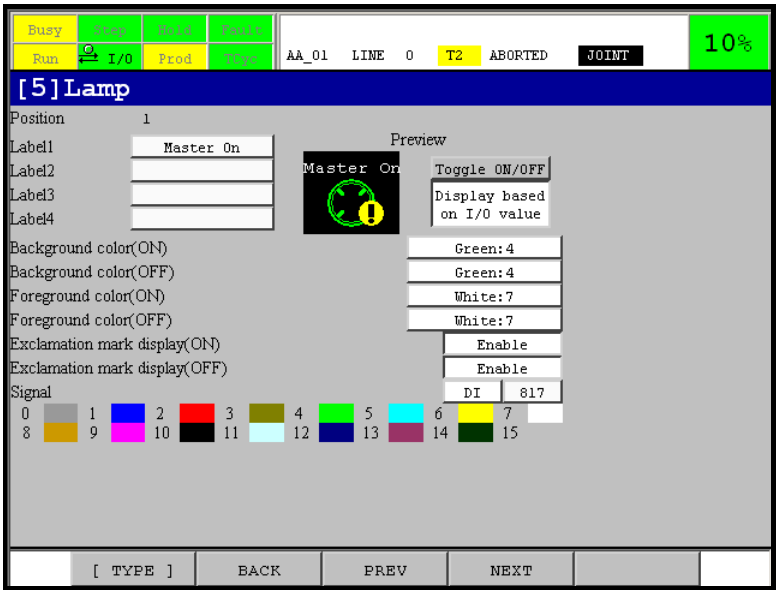
(1) It does not need to calculate compensation amount each time if the condition and motion is thoroughly same. However, it is needed to calculate compensation amount periodically because compensation amount changes depending on temperature and so on.
(2) It is needed to calculate compensation amount in the case where position is changed or approach direction is changed.
(3) Accuracy of position at TCP is influenced by tool rigidity. If tool rigidity is low, that might prevent to suppress the replacement of TCP.
(4) If offset_mode = 1, then this function does not work near singular points. In this case, ‘Convert offset fail’ is displayed on status window.

44 INTERFACE PANEL FUNCTION

This is a function to create panel similar to following figure. You can place buttons on grid like panel. The number of panels is 3 by default. You can create up to 10 panels. Please refer to Section 44.16 for change of the number of panels.



You can setup buttons by setup screen. Labels and signals for example.



Position 1
Label1 Master On
Label2
Label3
Label4
Background color(ON) Green:4
Background color(OFF) Green:4
Foreground color(ON) White:7
Foreground color(OFF) White:7
Exclamation mark display(ON) Enable
Exclamation mark display(OFF) Enable
Signal DI 817

You can put buttons on panel to display and change value of I/O.
Please refer to Section 44.77 ”TYPE OF BUTTON” for type of buttons.
This function can use button (which can be pressed) and lamp (which cannot be pressed).
In this document both button and lamp are often referred as button.

Hardware Requirement

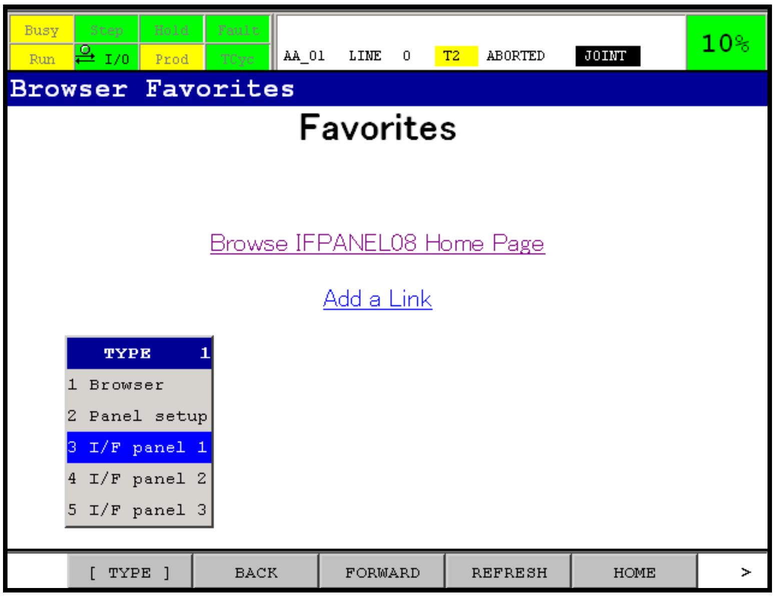
iPendant with touch panel is required.

Software Requirement

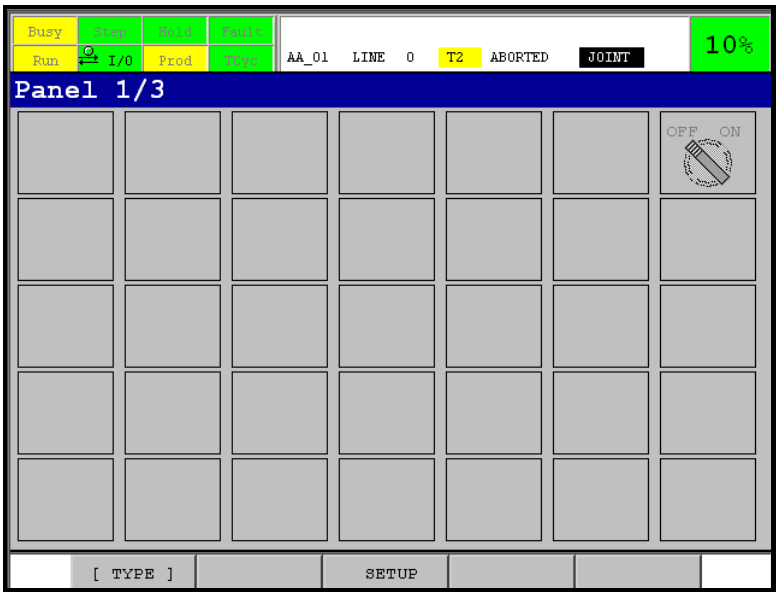
This function is optional (J741). This function is supported by system software version of 7DC2/02 or later.

44.1 HOW TO DISPLAY PANEL

1 Press [MENU] key.
2 Select BROWSER.
3 Press F1 [TYPE].
With default setting, you can see “I/F panel 1”, “I/F panel 2” and “I/F panel 3” in pull up menu. They are page 1, 2 and 3 of panel of this function. Select page to display.

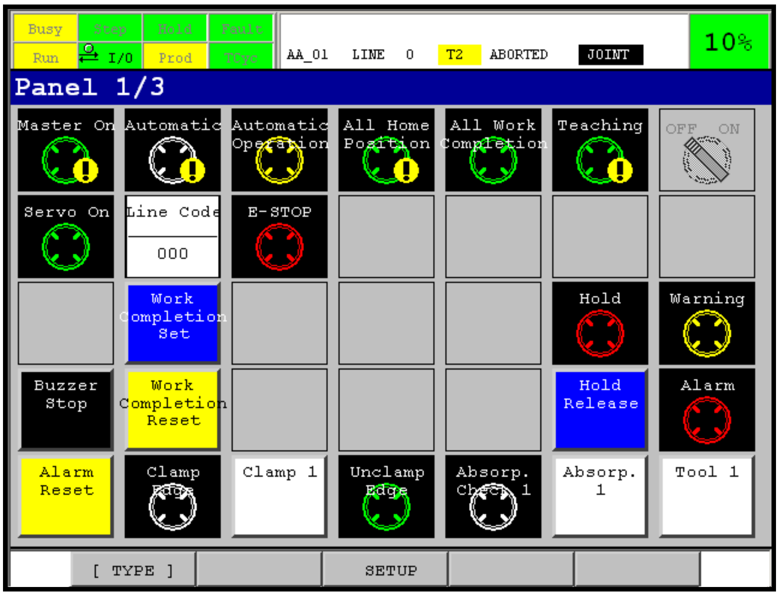


At first, there is no buttons on panels. You have to setup buttons to place them on panels.
Press F3 to display button type setup screen. You can select position you place button and type of button.
Properties of buttons are setup in button detail setup screen.
For button type setup screen and button detail setup screen, please refer to Section 44.4 and 44.6 respectively.



44.2 INTERFACE PANEL

Interface panel is operator's panel displayed on teach pendant. You can put lamps and buttons to display and turn ON/OFF I/O signal. Please refer to Section 44.7 for types of buttons.



This section explains elements that compose panel and condition to press button.

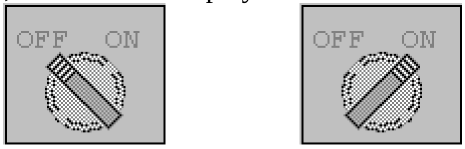
Page number

It is number displayed on title bar. In the figure above, title bar displays 1/3. It means this is the 1st page out of 3 pages.

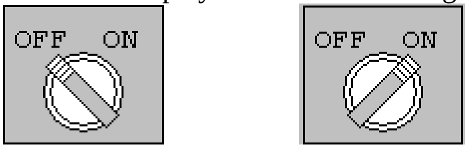
Enable switch

This is a button that is on the 1st row and the last column (right upper corner). By default setting, you have to turn on this button to press any button on panel.

When [SHIFT] key is not pressed, the button is displayed like the following figure.



When [SHIFT] key is pressed, the button is displayed like the following figure.



Panel enable switch is turned off when screen is changed. It is also true when page of pane is changed.

Conditions to press button

Condition to press button can be setup for each button. The condition is setup in operation condition setup screen. By default, following conditions have to be satisfied to press a button.

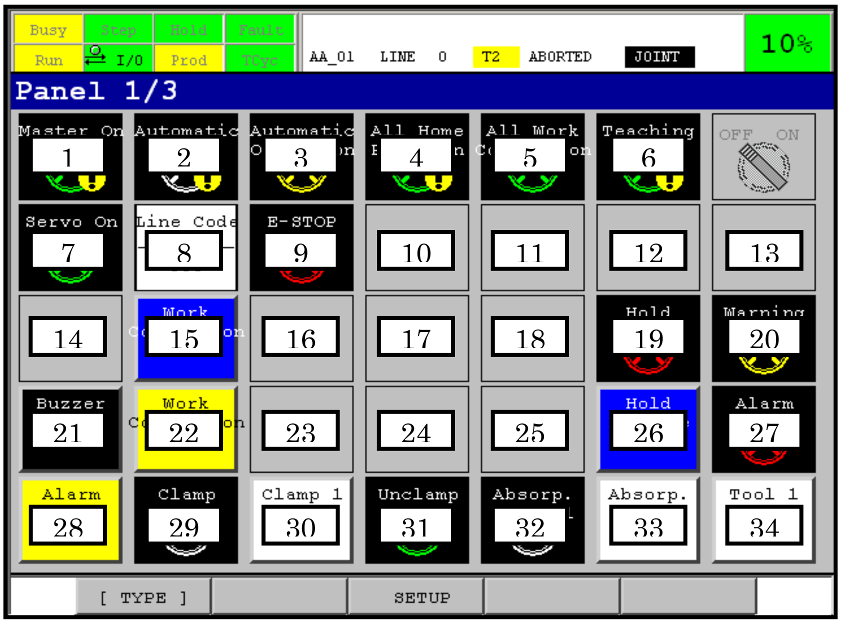
- Enable switch of Interface panel is ON
- Shift key is pressed.

Please refer to section 44.9, "OPERATION CONDITION" for detail of operation condition and its setup screen.

By default setting, operation condition of momentary button is checked only when it is pressed. Lamp does not have operation condition setup screen because it cannot be pressed.

Button

Interface panel is grid like panel. On each tile, button can be placed. Each tile has position number. Following figure shows position numbers of the 1st page of 7 column panel.

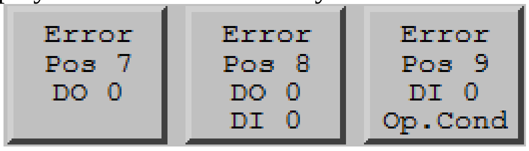


When you setup a button, you have to decide where to place it by selecting position number. Then type of the button on the position must be set. After the button is setup by detail screen, the button is displayed on panel on the position.

For setup, please refer to Section 44.3 ”INTERFACE PANEL SETUP SCREEN", 44.44 ” BUTTON TYPE SETUP SCREEN" and 44.5 ”SETTING OF TYPE OF BUTTON”.

Error button

If setup is wrong, button is displayed as error button. They are similar to the following figure.



Error
Pos 7
DO 0

Error
Pos 8
DO 0
DI 0

Error
Pos 9
DI 0
Op.Cond

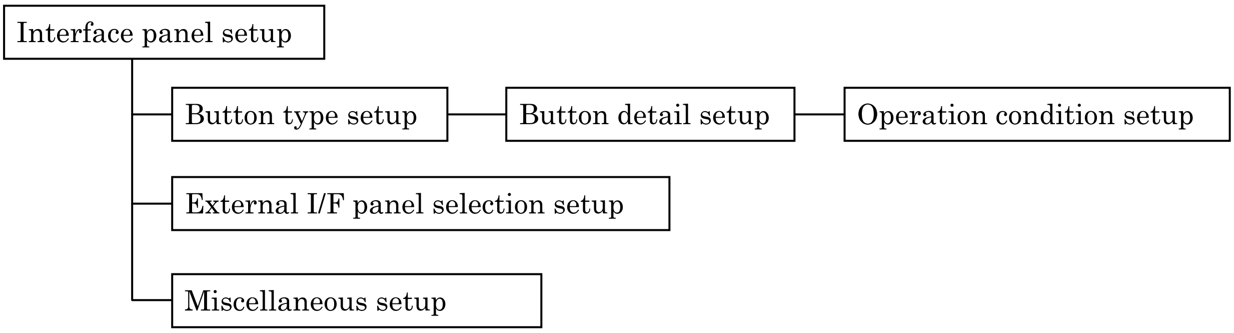
- The button displays position number and I/O that can not be read.
- If the button has 2 signals and both cannot be read, both signals are displayed.
  Button type 3 (2 contact point switch) displays only the 1st signal if index of the 2nd signal is 0.
- Button type 6 and 7 (digital switch and digital lamp) displays only the 1st found signal that cannot be read.
- If signal is specified and enabled as operation condition, and if the signal cannot be read, error button is displayed
  Signal for actual operation (output signal for example) is also wrong, it has priority.
- Press of the button displays detail setup screen.

F3 SETUP

Pressing the function key displays button type setup screen of the same page. For button type setup screen, please refer to section 44.4, “BUTTON TYPE SETUP SCREEN”.

44.3 INTERFACE PANEL SETUP SCREEN

Panel is setup by Interface panel setup screen. It consists of following screens.



Following table is overview of each screen.

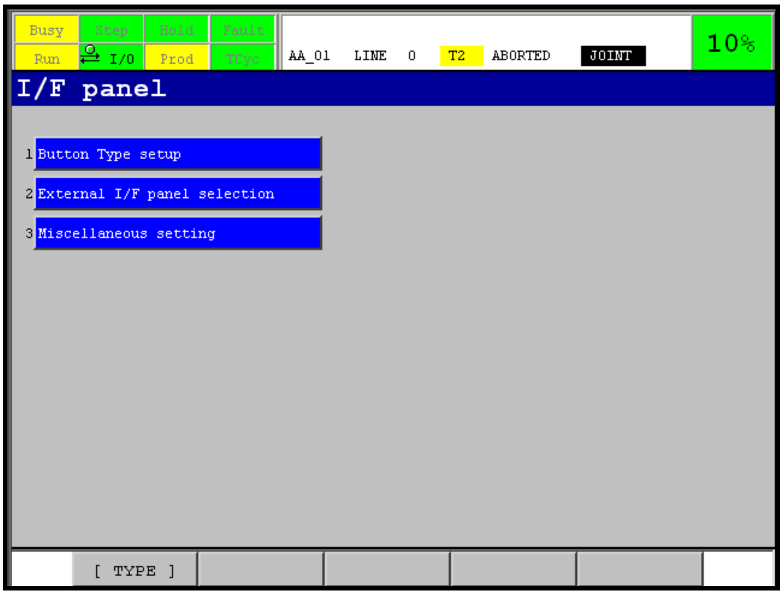
Screen	Description
Interface panel setup screen	The top level screen. This screen bundles the other screens in the table.
Button type setup screen	Type of buttons of each position number is set. You can copy and paste button in this screen.
External I/F panel selection setup screen	Digital input signals to display each page is setup
Misc. setup screen	Various setup items that affect all pages. The number of columns and pages are included.
Button detail setup screen	Various properties of buttons are set here.
Operation condition setup screen	This screen setup condition to press a button. This is part of button detail setup screen. Only buttons that can be pressed has this screen.

NOTE

Panel is setup for each language separately. When language is changed, setup for new language is used. Suppose that you setup panel in English only. If you change language to Japanese, there are no buttons on panels for Japanese. Setup in External I/F panel and Misc. setup screen don't depend on language.

You can display Interface panel setup screen by the following procedure.

1 Press [MENU] key.
2 Select SETUP.
3 Press F1 [TYPE].
4 Select I/F panel setup.
If this is the first visit to the screen since the last power up, you will see a screen similar to following figure.



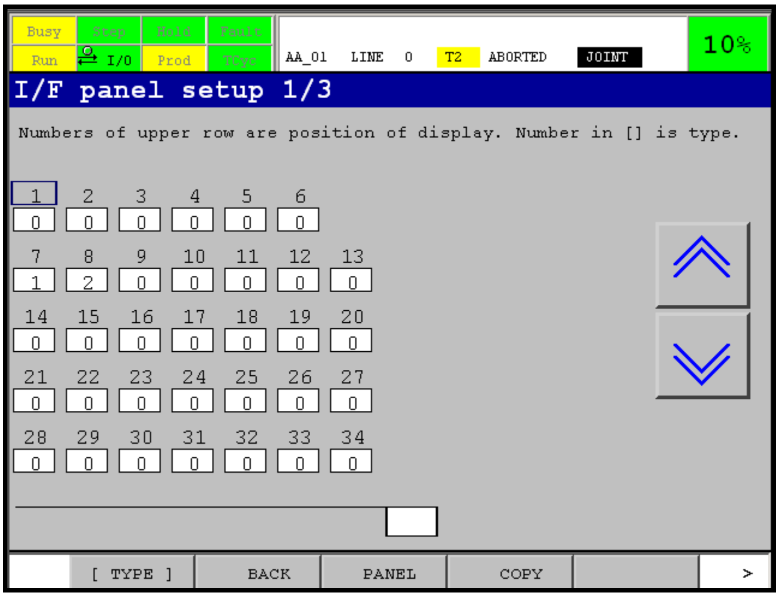
I/F panel

1 Button Type setup
2 External I/F panel selection
3 Miscellaneous setting

Press of each button leads you to each setup screen.

44.4 BUTTON TYPE SETUP SCREEN

This screen decides type of buttons. It is screen similar to the following figure.
Press of “Button type setup” button of Interface panel setup screen displays the screen.



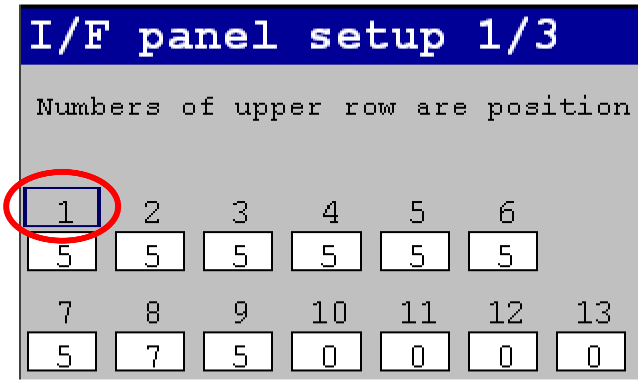
I/F panel setup 1/3

Numbers of upper row are position of display. Number in [] is type.

1	2	3	4	5	6	
0	0	0	0	0	0	
7	8	9	10	11	12	13
1	2	0	0	0	0	0
14	15	16	17	18	19	20
0	0	0	0	0	0	0
21	22	23	24	25	26	27
0	0	0	0	0	0	0
28	29	30	31	32	33	34
0	0	0	0	0	0	0

Numbers in white boxes show the type of button of the position. Number just above each white box is position number.
Press of arrows on the right side of the screen changes pages.
The arrows are displayed when cursor is on a position number.

Cursor



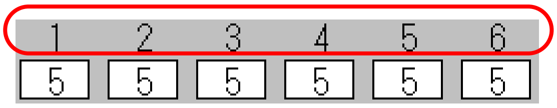
I/F panel setup 1/3

Numbers of upper row are position

1	2	3	4	5	6	
5	5	5	5	5	5	
7	8	9	10	11	12	13
5	7	5	0	0	0	0

In the figure above, blue rectangle that encloses “1” is cursor. You can move the cursor by arrow keys.

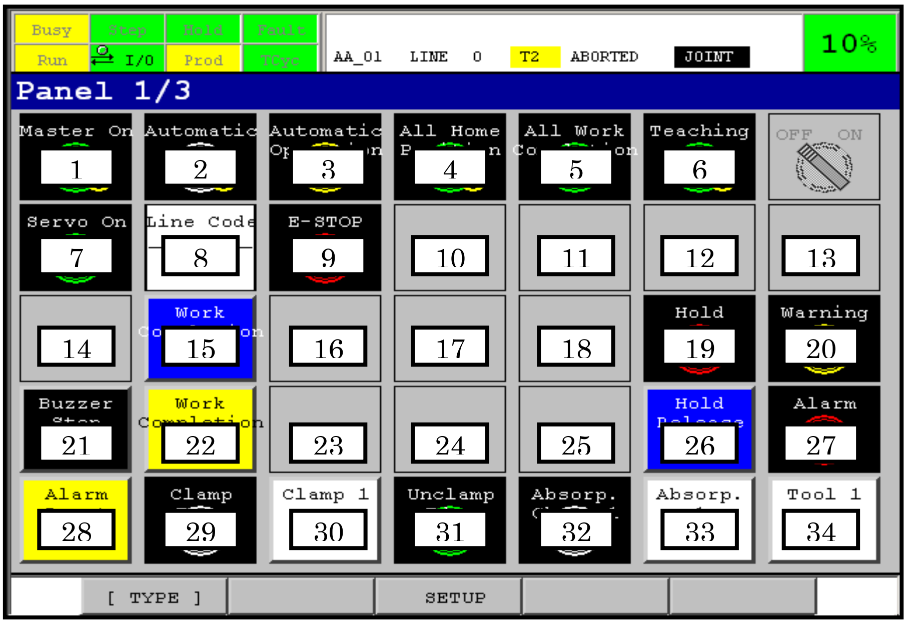
Position number



1	2	3	4	5	6
5	5	5	5	5	5

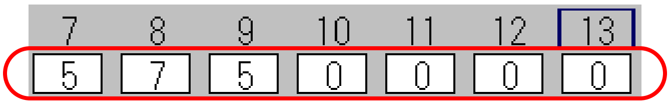
On the left side of the screen, there are many numbers. White box and the number above it are pair. Position numbers are the number above white boxes. When cursor is on a position number, arrow buttons to change page are displayed on the right side of screen. When cursor is on a position number, shift+ up arrow or down arrow key can be used to change pages.

Following figure shows position number of the 1st page of 7 column panel.



Likewise, left upper corner of the 2nd page is position number 35. Right lower corner corresponds to 68. You can display button detail setup screen if you press ENTER when cursor is on a position number.

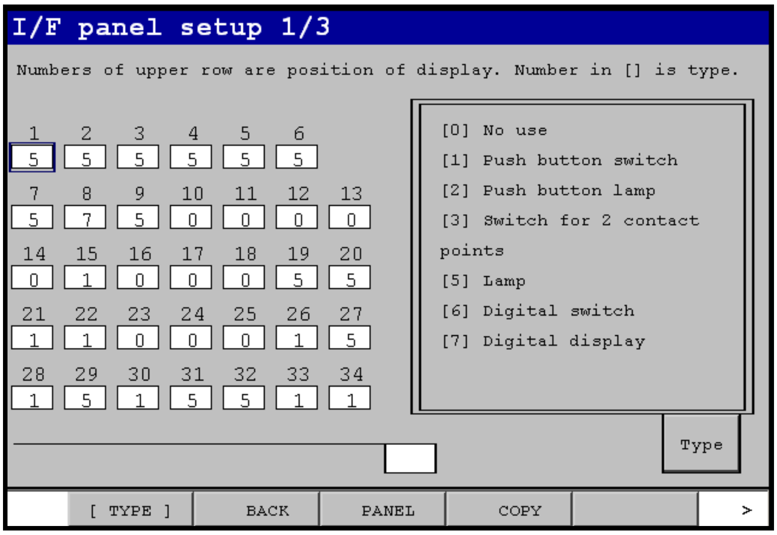
Type code field



7	8	9	10	11	12	13
5	7	5	0	0	0	0

This is white box on the left side of the screen. The box displays button type of corresponding position number, which is displayed above each white box. In the figure above, position 7 and 9 are type 5 (lamp). Position 8 is type 7 (digital lamp). From position 10 to 13 are type 0. It means the positions are not used.

When cursor is on type code field, explanation of type code is shown on the right side of the screen.



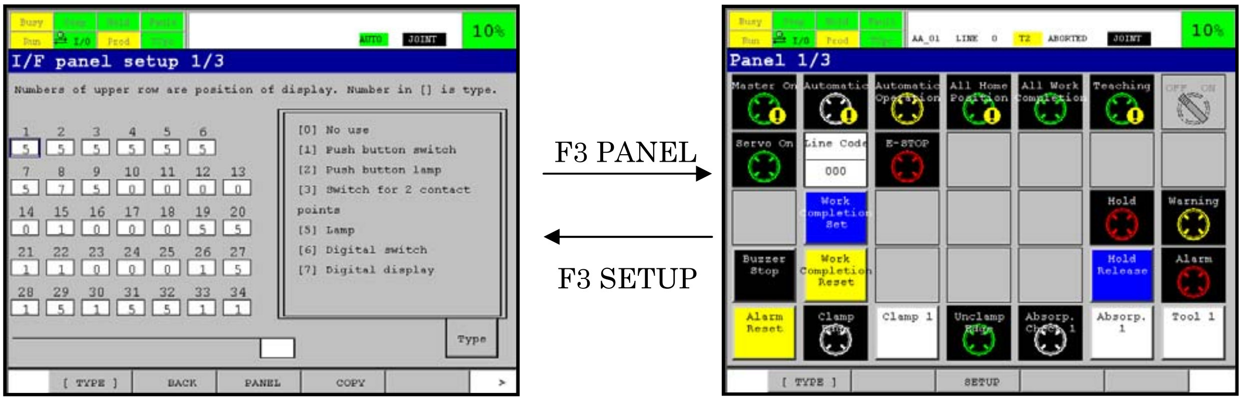
```
I/F panel setup 1/3
Numbers of upper row are position of display. Number in [] is type.

 1  2  3  4  5  6                [0] No use
 5  5  5  5  5  5                [1] Push button switch
                                 [2] Push button lamp
 7  8  9 10 11 12 13             [3] Switch for 2 contact
 5  7  5  0  0  0  0             points
                                 [5] Lamp
14 15 16 17 18 19 20             [6] Digital switch
 0  1  0  0  0  5  5             [7] Digital display

21 22 23 24 25 26 27
 1  1  0  0  0  1  5

28 29 30 31 32 33 34
 1  5  1  5  5  1  1
                                                     Type

 [ TYPE ]   BACK   PANEL   COPY                        >
```

F2 BACK

The highest level screen, Interface panel setup screen is displayed.

F3 PANEL

By pressing F3 PANEL, panel of the same page is displayed.



F3 SETUP of panel displays back to button type setup screen of the same page.
Even in this case, F2 BACK always displays interface panel setup screen (the highest level screen).

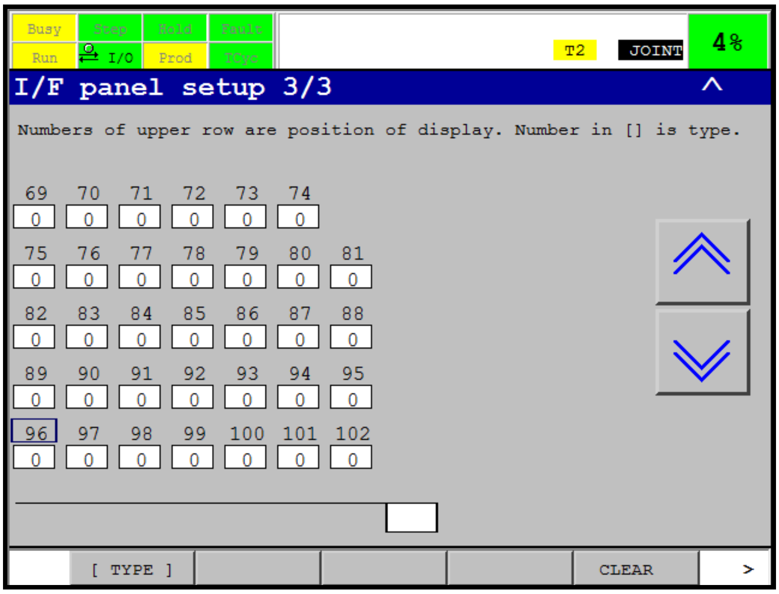
F4 COPY

You can use this function key to copy a button. Please refer to Subsection 44.5.1 for more details.

44.5 SETTING OF TYPE OF BUTTON

Type of button is setup by following procedure.

1 Display button type setup screen.
2 Display page you want to setup.
3 Move cursor to position number or type code field (white box) of the position.

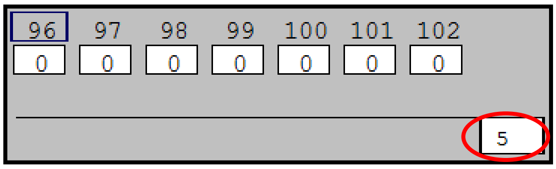


I/F panel setup 3/3

Numbers of upper row are position of display. Number in [] is type.

```
69  70  71  72  73  74
 0   0   0   0   0   0
75  76  77  78  79  80  81
 0   0   0   0   0   0   0
82  83  84  85  86  87  88
 0   0   0   0   0   0   0
89  90  91  92  93  94  95
 0   0   0   0   0   0   0
96  97  98  99  100 101 102
 0   0   0   0   0   0   0
```

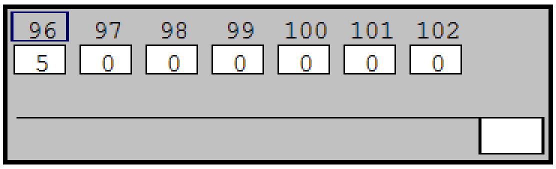
4 Input type code.



```
96  97  98  99  100 101 102
 0   0   0   0   0   0   0
                          5
```

Type code you are inputting is displayed in white box at the bottom of the screen.

5 Press ENTER to complete input.
It is reflected to type code field of the position you selected.

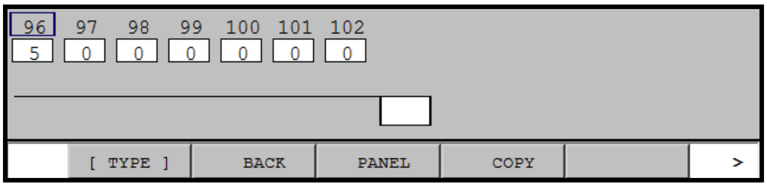


```
96  97  98  99  100 101 102
 5   0   0   0   0   0   0
```

44.5.1 Copy and Paste

You can copy and paste a button in button type setup screen.

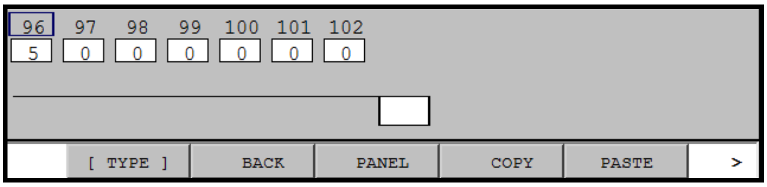
1 Display button type setup screen.
2 Display the page that includes button to copy.
3 Move cursor to position number or type code field of the button.



```
96  97  98  99  100 101 102
 5   0   0   0   0   0   0
```

In the figure above, the button of position 96 (type 5, lamp) is copied

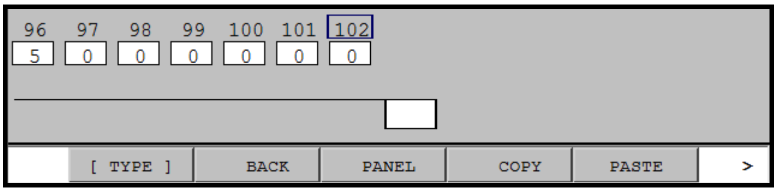
4 Press F4, COPY.



96	97	98	99	100	101	102
5	0	0	0	0	0	0

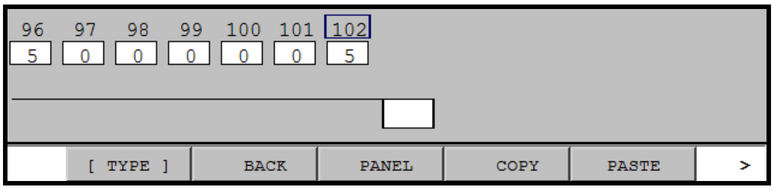
F5 PASTE is displayed. If you have copied another button before, PASTE must be already displayed before you press F4 copy. Copy data is overwritten to previous data.

5 Move cursor to position to paste.



96	97	98	99	100	101	102
5	0	0	0	0	0	0

6 Press F5, PASTE.

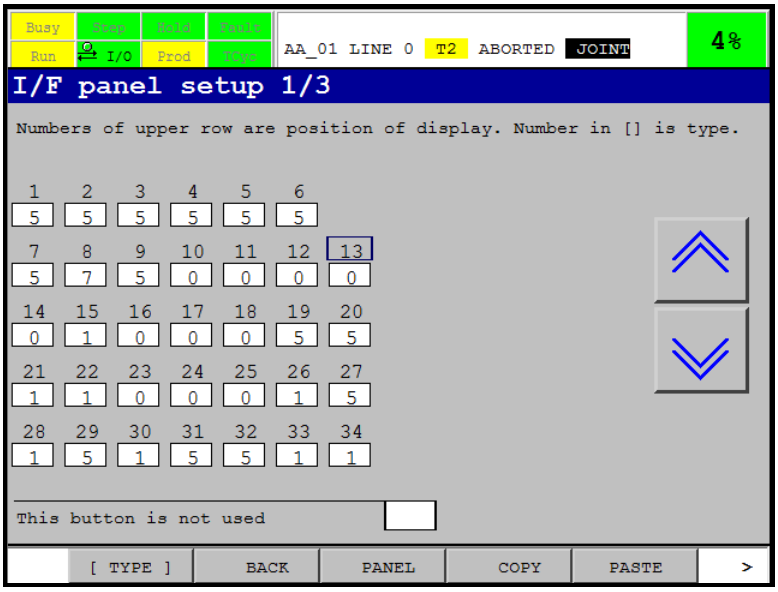


96	97	98	99	100	101	102
5	0	0	0	0	0	5

The button was copied. You can paste the same data repeatedly.

NOTE

You cannot copy button of type 0 (not used). Warning message is displayed as following figure. When you set type code to 0, input 0 one by one, confirming if the button can be cleared.



I/F panel setup 1/3

Numbers of upper row are position of display. Number in [] is type.

1	2	3	4	5	6	
5	5	5	5	5	5	
7	8	9	10	11	12	13
5	7	5	0	0	0	0
14	15	16	17	18	19	20
0	1	0	0	0	5	5
21	22	23	24	25	26	27
1	1	0	0	0	1	5
28	29	30	31	32	33	34
1	5	1	5	5	1	1

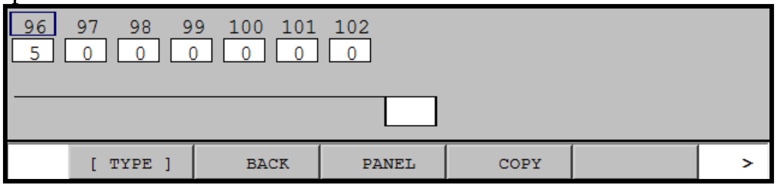
This button is not used

You cannot paste if current language is different from the one selected at copy. Suppose that you copied a button when English was selected. You cannot paste the data if the other language is selected.

44.6 BUTTON DETAIL SETUP SCREEN

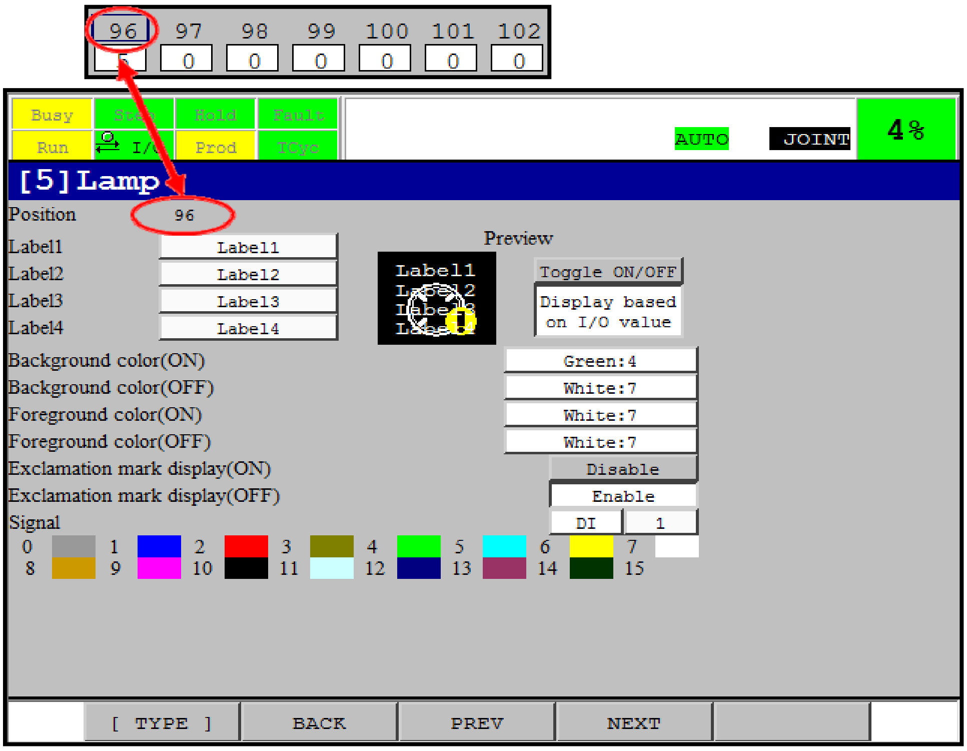
Button detail setup screen is a screen to setup various property of a button. To display the screen please use following procedure.

1 Display button type setup screen.
2 Change pages to display panel that includes the button to be setup.
3 Move cursor to position number of the button.



96	97	98	99	100	101	102
5	0	0	0	0	0	0

4 Press ENTER.
Screen for the button on the position (position number) is displayed. According to type of button, proper screen is displayed. In the following figure, screen for type 5 (lamp) is displayed.



96	97	98	99	100	101	102
5	0	0	0	0	0	0

[5]Lamp

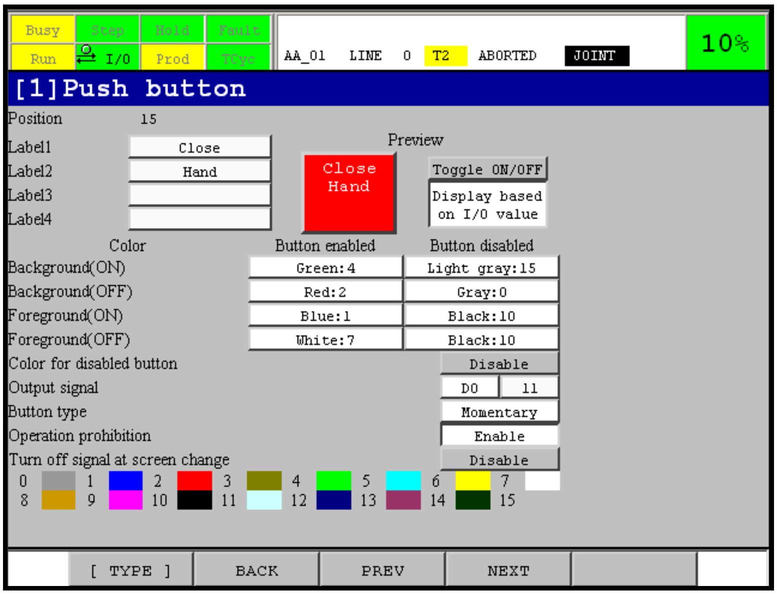
Position 96
Label1 Label1
Label2 Label2
Label3 Label3
Label4 Label4

Background color(ON) Green:4
Background color(OFF) White:7
Foreground color(ON) White:7
Foreground color(OFF) White:7
Exclamation mark display(ON) Disable
Exclamation mark display(OFF) Enable
Signal DI 1

NOTE
After change of button type, index of input or output signals are defaulted to 0.
Preview of button always shows error.

44.6.1 Preview of Button

Button detail setup screen has preview of button. You can get a rough idea how the button actually looks on panel. Some items cannot be reflected to preview. They are mainly items to decide action of button when it is pressed. This subsection explains overview of preview.



Busy Step Hold Fault
Run I/O Prod TCyc
AA_01 LINE 0 T2 ABORTED JOINT 10%

[1]Push button

Position 15

		Preview	
Label1	Close	Close Hand	Toggle ON/OFF
Label2	Hand		Display based on I/O value
Label3			
Label4			

Color	Button enabled	Button disabled
Background(ON)	Green:4	Light gray:15
Background(OFF)	Red:2	Gray:0
Foreground(ON)	Blue:1	Black:10
Foreground(OFF)	White:7	Black:10
Color for disabled button		Disable
Output signal		DO 11
Button type		Momentary
Operation prohibition		Enable
Turn off signal at screen change		Disable

0 1 2 3 4 5 6 7
8 9 10 11 12 13 14 15

[TYPE] BACK PREV NEXT

“Toggle ON/OFF” button

The button exists in detail screen of buttons that has ON/OFF status. They have colors for ON and OFF state.

The button toggles “ON” state and “OFF” state of button without changing actual signal the button should refer to. The button works only when “Display based on I/O value” button is OFF.

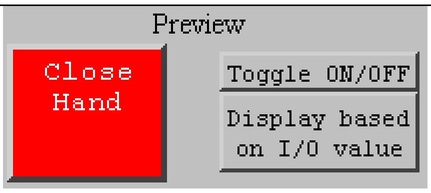
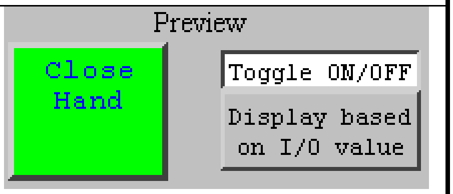
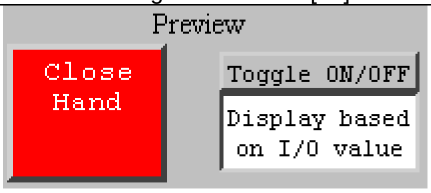
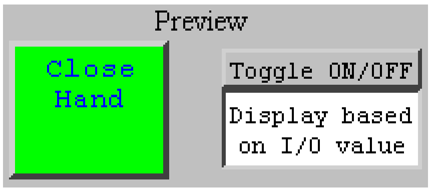
In case of 2 contact switch (type 3), left/right or up/down state are toggled.

“Display based on I/O value” button

When the button is ON, preview is displayed based on current status of I/O specified. After this button is pressed, “Toggle ON/OFF” button is turned off automatically.

Based on the setup in the previous figure, we explain how the button works. DO [11] is specified as output signal. When panel is actually displayed, button is displayed based on DO [11].

The following table shows how “Toggle ON/OFF” and “Display based on I/O value” button change preview.

“Display based on I/O value”	“Toggle ON/OFF” button” is OFF	“Toggle ON/OFF” button” is ON
OFF (“Toggle ON/OFF” button decides status of preview button.)	 "OFF" state regardless of DO[11]	 "ON" state regardless of DO[11]
ON	  Status of DO [11] decides status of preview button.	This status doesn’t exist. You cannot turn on both “Toggle ON/OFF” button and “Display based on I/O value” at the same time. If you press one of them when the other is ON, the latter is turned off automatically.

NOTE

You cannot press a button displayed as preview of button detail setup screen. The button doesn't respond.

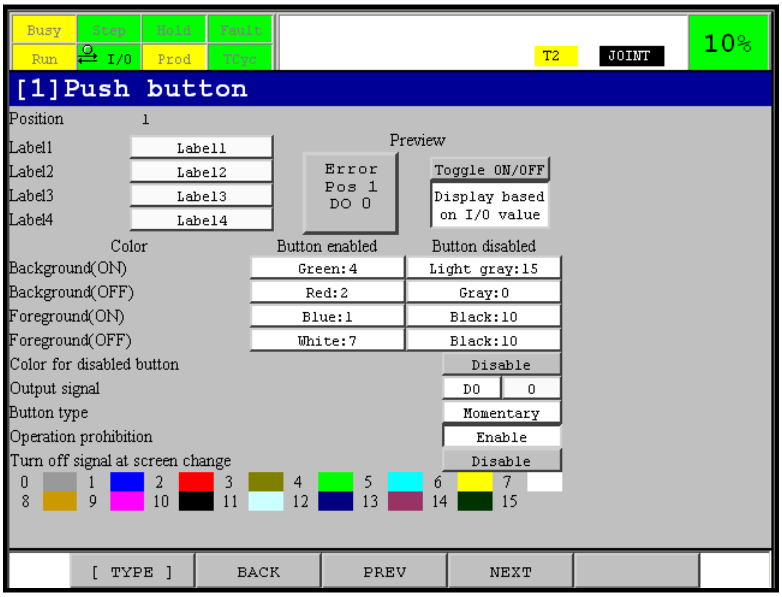
Color for disabled button

A few types of buttons have setup item, “Color for disabled button”. If the item is set to “Enable”, color of button is changed according to whether condition to press button (operation condition) is satisfied or not. You can compare the two looks by preview. In preview, panel enable switch is treated as if it is always ON in operation condition check.

When "Color for disabled button" is enabled and operation condition is default setting, preview button is displayed in color of “Button enabled” column when [SHIFT] key is pressed.

44.6.2 Error Display of Button

When “Display based on I/O” is pressed and specified signal cannot be read, preview of detail screen displays error button.



[1]Push button

Position 1

Label1 Label1

Label2 Label2

Label3 Label3

Label4 Label4

Color	Button enabled	Button disabled
Background(ON)	Green:4	Light gray:15
Background(OFF)	Red:2	Gray:0
Foreground(ON)	Blue:1	Black:10
Foreground(OFF)	White:7	Black:10

Color for disabled button Disable

Output signal DO 0

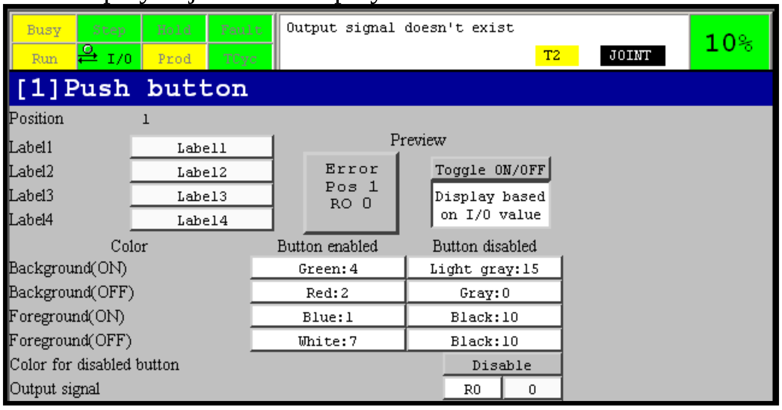
Button type Momentary

Operation prohibition Enable

Turn off signal at screen change Disable

When “Display based on I/O” button is not pressed, preview is not displayed as error button.

When setup item is changed but the changed item still causes error, message is displayed on error line. This error message is not displayed just after display of detail screen.



Output signal doesn't exist

[1]Push button

Position 1

Label1 Label1

Label2 Label2

Label3 Label3

Label4 Label4

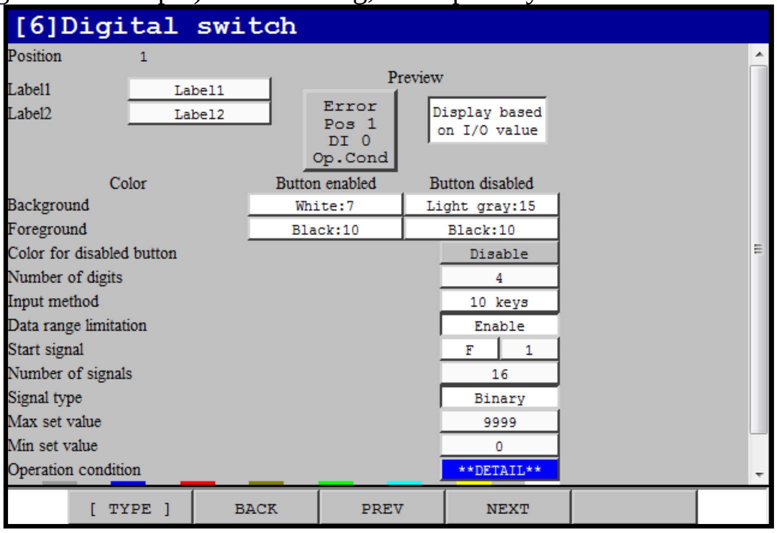
Color	Button enabled	Button disabled
Background(ON)	Green:4	Light gray:15
Background(OFF)	Red:2	Gray:0
Foreground(ON)	Blue:1	Black:10
Foreground(OFF)	White:7	Black:10

Color for disabled button Disable

Output signal RO 0

When setup item is changed successfully, the error message is cleared. In the above figure, any color change would clear error message on error line. However, preview still shows error button.

When operation condition signal cannot be read, preview displays error button. Signal for actual operation (output signal for example) is also wrong, it has priority.



[6]Digital switch

Position 1

Label1 Label1

Label2 Label2

Preview

Error
Pos 1
DI 0
Op.Cond

Display based on I/O value

Color	Button enabled	Button disabled
Background	White:7	Light gray:15
Foreground	Black:10	Black:10
Color for disabled button		Disable
Number of digits		4
Input method		10 keys
Data range limitation		Enable
Start signal		F 1
Number of signals		16
Signal type		Binary
Max set value		9999
Min set value		0
Operation condition		**DETAIL**

In the figure above, error button shows DI 0 is wrong. But DI[0] is not specified in the screen. It is specified in operation condition setup screen. When error button is displayed, please check operation condition screen, too.

44.7 TYPE OF BUTTON

This function can use following button and lamp.

- Push button (type 1)
- Push button lamp (type 2)
- 2 contact points switch (type 3)
- Lamp (type 5)
- Digital switch (type 6)
- Digital lamp (type7)

Overview of properties that is common to some button types are explained in the next section.

44.8 COMMON PROPERTIES

Label

This is string shown on button. The number of labels depends on type of button.

They are shown as Label1,…, Label4 in button detail setup screen. Label n is displayed at the nth row of the button.

When panel has 7 columns, each label should have less than 10 characters.

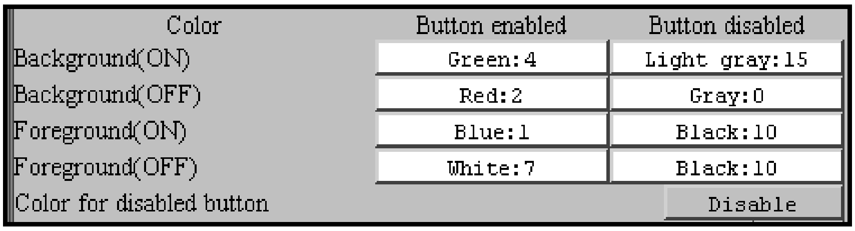
If the number of column is 8, less than 8 characters.

Foreground color and background color

Following type of buttons have two sets of colors. “Button enabled” and “Button disabled” column of button detail setup screen.

- Push button (type 1)
- Push button lamp (type 2)
- 2 contact points switch (type 3)
- Digital switch (type 6)

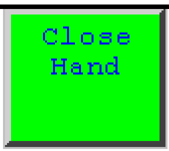
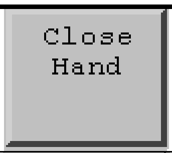
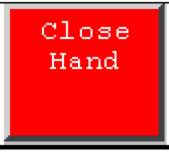
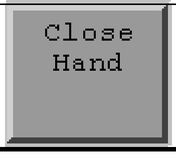
Following figure is extract of button detail setup screen of type 1, 2 and 3.



Color	Button enabled	Button disabled
Background(ON)	Green:4	Light gray:15
Background(OFF)	Red:2	Gray:0
Foreground(ON)	Blue:1	Black:10
Foreground(OFF)	White:7	Black:10
Color for disabled button		Disable

Digital switch doesn't have ON/OFF status. There are only Foreground, Background and color for disabled button.
Colors of "Button enabled" column are for when button can be pressed (when button is enabled).
"Button disabled" column is for when button can NOT be pressed (when button is disabled).
Please refer to section 44.9 "OPERATION CONDITION" for conditions to press a button.

Item	Description
Background(ON)	Background color for when signal is ON. It depends on button type which signal decides color. Left column is for when button is enabled (when button is enabled). Right column is for when button is disabled.
Background(OFF)	Background color for when signal is OFF.
Foreground(ON)	Foreground color for when signal is ON.
Foreground(OFF)	Foreground color for when signal is OFF.
Color for disabled button	Enable Color of button Is changed according to whether button can be pressed or not. Disable Colors of "Button enabled" column are always used.

Signal	Button is enabled	Button is disabled
ON		
OFF		

NOTE
If "Color for disabled button" is disabled, colors of "Button enabled" column are always used.

Turn off signal at screen change

If this item is enabled, specified output is turned off at following event

- When panel is displayed
- When screen is changed to another screen (This includes page change between panels)
- Any operation that re-displays panel (Refresh pane of FCTN menu, Zoom for example)

Supported signal type

Buttons of this function supports digital type I/O only. I/O that has integer value are not supported. For example, analog I/O and group I/O are not supported. Digital switch and lamp treat a series of digital signal as a group and treat them as numeric value.
Following I/O types are supported by output signal of push button (type 1), output signal of push button lamp (type 2), both signals of 2 points contact switch (type 3) and signals of digital switch (type 6).

- DO

- RO
- F (Flag)

Status signal of push button lamp (type 2), signal of lamp (type 5) and signals of digital lamp (type 7) support following I/O types.

- DI
- DO
- RI
- RO
- SI
- SO
- TPOUT
- UI
- UO
- F (Flag)
- M (Marker)

TPOUT[1-8] corresponds to status of software LED, which is at left upper corner of iPendant screen. For software LED, please refer to “Teach Pendant” subsection of R-30*i*B/R-30*i*B CONTROLLER OPERATOR’S MANUAL (Basic Operation) (B-83284EN).

For Flag and Marker, please refer to section of “BACKGROUND LOGIC” of R-30*i*B/R-30*i*B CONTROLLER OPERATOR’S MANUAL (Basic Operation). Flag and Marker are explained in subsection of “Other Instructions and Functions”.

Operation condition

This item exists in detail screen of button type that can be pressed. You can display operation condition setup screen by pressing detail button on the row of operation condition. Please refer to section 44.9 OPERATION CONDITION for more information.

44.9 OPERATION CONDITION

Interface panel function can specify condition to press buttons on panel.

- Conditions are specified by combining following keys and I/O.
  - Panel enable switch
  - [SHIFT] key
  - TP enable switch
  - DEADMAN switch
  - Signal specified by user is ON or OFF
- Conditions are specified for each button.
- If a button is pressed when its operation condition is not satisfied, warning is posted.
  Some button may look the button was successfully pressed, but the function of the button does not work. Output signal is not turned on for example.

Signal specified by operation condition setup screen is referred as operation condition signal.

44.9.1 Button Types Supported

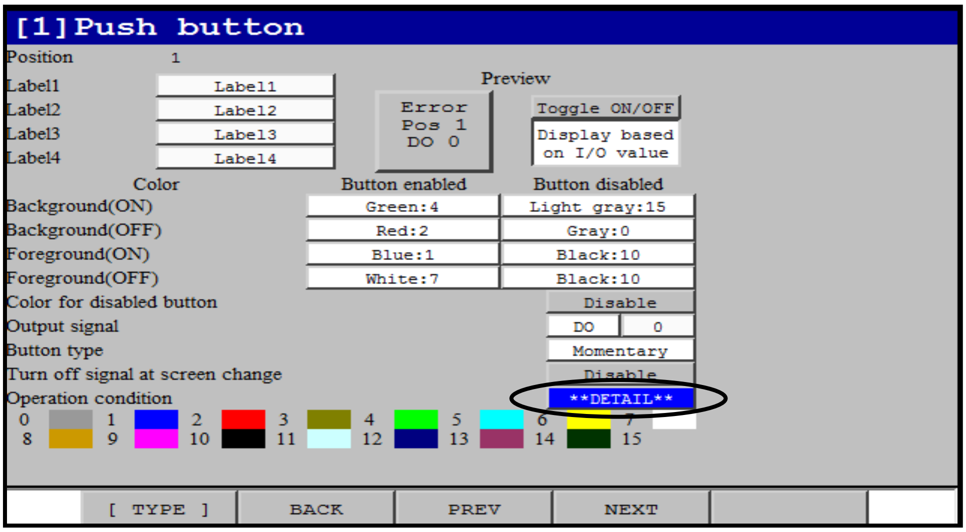
Operation condition is supported by following button types.

- Push button (type 1)
- Push button lamp (type 2)
- 2 contact point switch (type 3)
- Digital switch (type 6)

Lamp (type 5) and Digital lamp (type 7) don’t support operation condition. They are lamp and cannot be pressed.

44.9.2 Operation Condition Setup Screen

The screen is displayed by “Detail” button of button detail setup screen.



[1]Push button

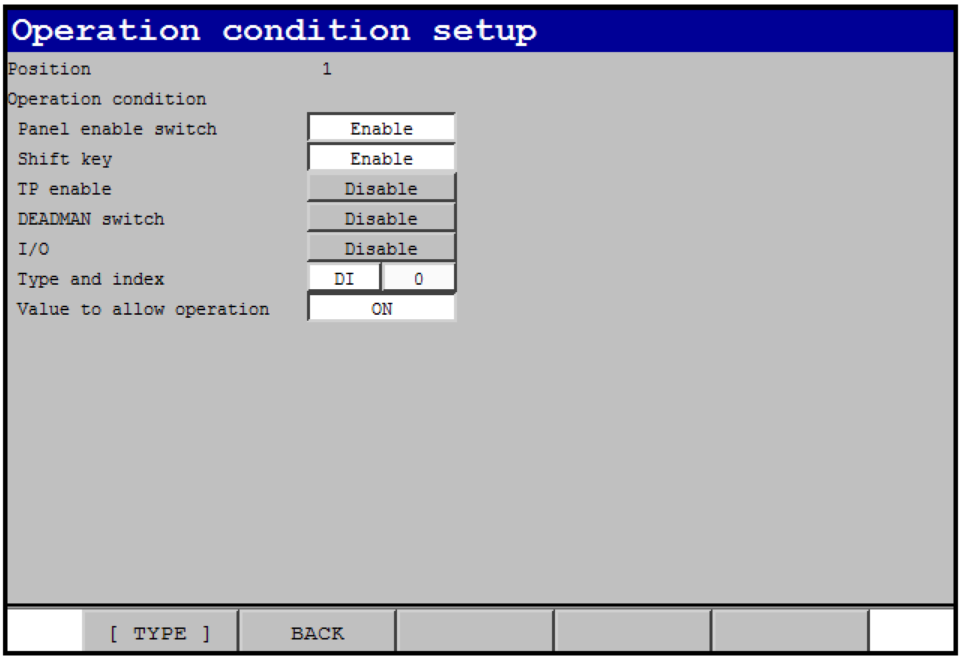
Position 1

Label1	Label1	
Label2	Label2	
Label3	Label3	
Label4	Label4	

Preview: Error Pos 1 DO 0 / Toggle ON/OFF / Display based on I/O value

Color	Button enabled	Button disabled
Background(ON)	Green:4	Light gray:15
Background(OFF)	Red:2	Gray:0
Foreground(ON)	Blue:1	Black:10
Foreground(OFF)	White:7	Black:10

Color for disabled button: Disable
Output signal: DO 0
Button type: Momentary
Turn off signal at screen change: Disable
Operation condition: **DETAIL**



Operation condition setup

Position 1

Operation condition

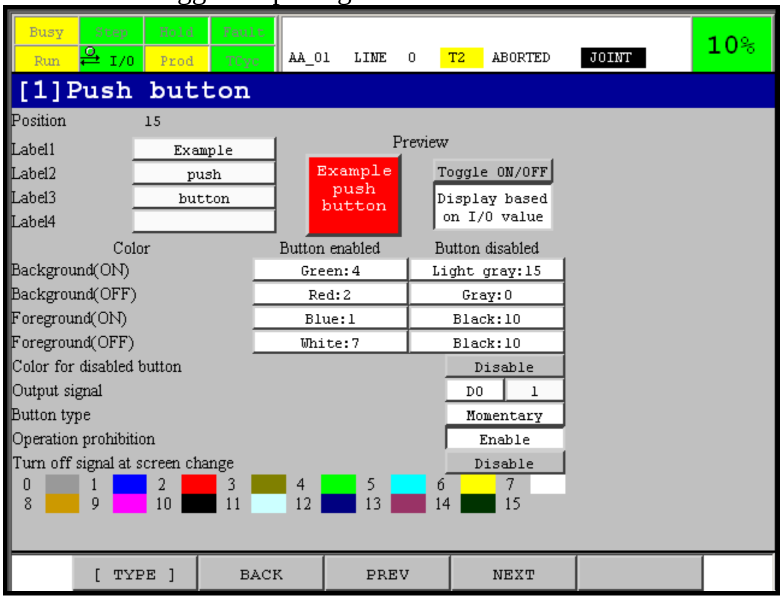
Panel enable switch	Enable
Shift key	Enable
TP enable	Disable
DEADMAN switch	Disable
I/O	Disable
Type and index	DI 0
Value to allow operation	ON

Item	Default value	Description
Position	Display only	Position number of button under setup
Panel enable switch	Enable	Enable Button cannot be pressed if panel enable switch is OFF. If pressed, “IFPN-010 Enable Interface panel” is posted. Disable Status of panel enable switch does not affect whether button can be pressed or not.
SHIFT key	Enable	Enable Button cannot be pressed if shit key is not pressed. If the button is pressed, “IFPN-009 Press [SHIFT] key” is posted. Disable Status of [SHIFT] key does not affect whether button can be pressed or not.

Item	Default value	Description
TP enable	Disable	Enable Button cannot be pressed if TP is disabled. If pressed, “IFPN-019 TP is disabled” is posted. Disable Status of TP enable switch does not affect whether button can be pressed or not.
DEADMAN switch	Disable	Enable Following conditions must be satisfied to press the button. - TP E-Stop button is OFF - TP is enabled - TP DEADMAN switch is gripped properly Otherwise, one of following warning is posted when the button is pressed. - IFPN-018 TP Estop is pressed - IFPN-019 TP is disabled - IFPN-020 Deadman switch is not gripped properly Disable The conditions above aren’t checked.
I/O	Disable	Enable At press of button, value of signal specified by item “Type and index” (Operation condition signal) is checked. If the value isn’t equal to value specified by item “Value to allow operation”, button cannot be pressed. If pressed, “IFPN-022 Condition signal is OFF” or “IFPN-023 Condition signal is ON” is posted. Disable Signal specified by “Type and index” is not checked.
Type and index	DI and 0	This item is used only when item “I/O” is enabled. Signal to check at press (Operation condition signal) is specified.
Value to allow operation	ON	This item is used only when item “I/O” is enabled. This item decides operation condition signal should be ON or OFF to press the button.

44.10 PUSH BUTTON

Type code is 1. This is button to toggle output signal.



[1]Push button

Position	15	
Label1	Example	
Label2	push	
Label3	button	
Label4		
Color	Button enabled	Button disabled
Background(ON)	Green:4	Light gray:15
Background(OFF)	Red:2	Gray:0
Foreground(ON)	Blue:1	Black:10
Foreground(OFF)	White:7	Black:10
Color for disabled button		Disable
Output signal		DO 1
Button type		Momentary
Operation prohibition		Enable
Turn off signal at screen change		Disable

0 ▨ 1 🟦 2 🟥 3 ▨ 4 🟩 5 ▨ 6 🟨 7 □
8 ▨ 9 ▨ 10 ■ 11 ▨ 12 ▨ 13 ▨ 14 ▨ 15 ▨

Following table is list of setup items.

Two default values are written in default value column of color items. The 1st one is value for "Button enabled" column of detail setup screen. The 2nd one is for "Button disabled column".

Please refer to "Foreground color and background color" of section 44.8 "COMMON PROPERTIES" for more information of setup of the colors.

Item	Default value	Description
Label1, …, Label4	Label1,…,Label4	These are strings displayed on the button.
Background (ON)	Green(4) and Light gray(15)	Background color of when output signal is ON.
Background (OFF)	Red(2) and Gray(0)	Background color of when output signal is OFF.
Foreground (ON)	Blue(1) and Black(10)	Foreground color of when output signal is ON. This is color of label.
Foreground (OFF)	White (7) and Black(10)	Foreground color of when output signal is OFF
Color for disabled button	Disable	When this item is enabled, color of "Button disabled" column is used when the button is disabled. Please refer to section 44.8 "COMMON PROPERTIES" for details.
Output signal	DO[0]	Output signal to be toggled by this button.
Button type	Momentary	This item decides how button operates. Choices are "Momentary" or "Alternate". Momentary When button is pressed, the button turns on. When button is released, the button turns off. Alternate Every time the button is pressed, the button turns on and off.
Turn off signal at screen change	Disable	If this item is enabled, specified output is turned off at a few events. Please refer to 44.8 "COMMON PROPERTIES"
Operation condition		"Detail" button displays operation condition screen.

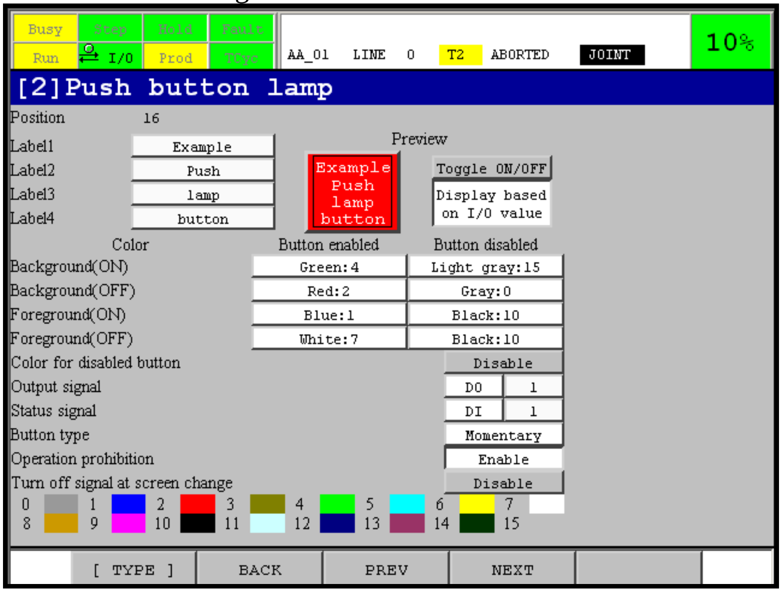
NOTE

If button type is momentary (default), specified output signal turns off when;

- The panel is displayed.
- Screen is changed to another one, including another page of panel.
- Any operation that re-displays panel
  (Refresh pane of FCTN menu, Zoom for example)

44.11 PUSH BUTTON LAMP

Type code is 2. This is push button to turn on and off output signal. Color is changed based on another signal. The signal is referred as status signal.



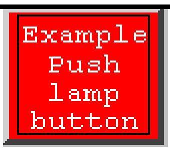
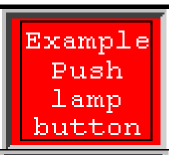
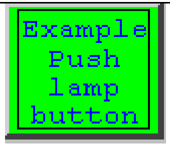
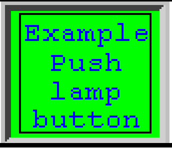
[2]Push button lamp

Position 16

Label1	Example	Preview
Label2	Push	Toggle ON/OFF
Label3	lamp	Display based on I/O value
Label4	button	

Color	Button enabled	Button disabled
Background(ON)	Green:4	Light gray:15
Background(OFF)	Red:2	Gray:0
Foreground(ON)	Blue:1	Black:10
Foreground(OFF)	White:7	Black:10
Color for disabled button		Disable
Output signal		DO 1
Status signal		DI 1
Button type		Momentary
Operation prohibition		Enable
Turn off signal at screen change		Disable

Foreground and background color is decided by status signal (input signal). If button type is alternate, button is pressed state when output signal is on. Button is released state when output signal is off.

Status signal	Released state	Pressed state
OFF		
ON		

Following table is list of setup items.

Two default values are written in default value column of color items. The 1st one is value for "Button enabled" column of detail setup screen. The 2nd one is for "Button disabled" column.

Please refer to "Foreground color and background color" of section 44.8 "COMMON PROPERTIES" for more information of setup of the colors.

Item	Default value	Description
Label1, …, Label4	Label1,…,Label4	These are strings displayed on the button.
Background (ON)	Green(4) and Light gray(15)	Background color of when status signal is ON. This is similar to Background (ON) of push button lamp but color doesn't depend on output signal. Status signal is referred to decide color.
Background (OFF)	Red(2) and Gray(0)	Background color of when status signal is OFF.
Foreground (ON)	Blue(1) and Black(10)	Foreground color of when status signal is ON. This is color of label.
Foreground (OFF)	White (7) and Black(10)	Foreground color of when status signal is OFF.

Item	Default value	Description
Color for disabled button	Disable	When this item is enabled, color of “Button disabled” column is used when the button is disabled. Please refer to section 44.8 "COMMON PROPERTIES” for details.
Output signal	DO[0]	Output signal to be toggled by this button.
Status signal	DI[0]	Color of this button changes according to value of this signal.
Button type	Momentary	This item decides how button operates. Please refer to explanation for same item of push button lamp.
Turn off signal at screen change	Disable	Please refer to explanation for the same item of push button lamp.
Operation condition		“Detail” button displays operation condition screen.

NOTE

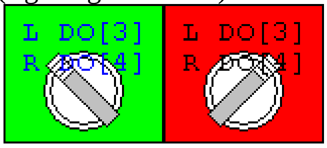
If button type is momentary (default), specified output signal turns off when;

- The panel is displayed.
- Screen is changed to another one, including another page of panel.
- Any operation that re-displays panel
  (Refresh pane of FCTN menu, Zoom for example)

44.12 2 CONTACT POINT SWITCH

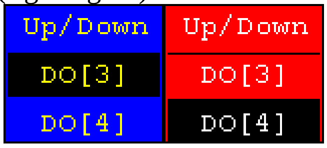
Type code is 3. This is switch to toggle status of 2 output signals.
Suppose DO [3] and DO [4] are specified as signals of 2 contact point switch.
The button can toggle following 2 status

- DO [3] is ON and DO [4] is OFF (left figure below)
- DO [3] is OFF and DO [4] is ON (right figure below)

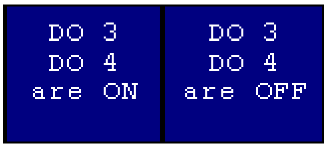


This type of button has another display format, up-down. If you use up-down format, the number of labels are 3. Following figures are example of up-down format with same signals specified.

- DO [3] is ON and DO [4] is OFF (left figure)
- DO [3] is OFF and DO [4] is ON (right figure)



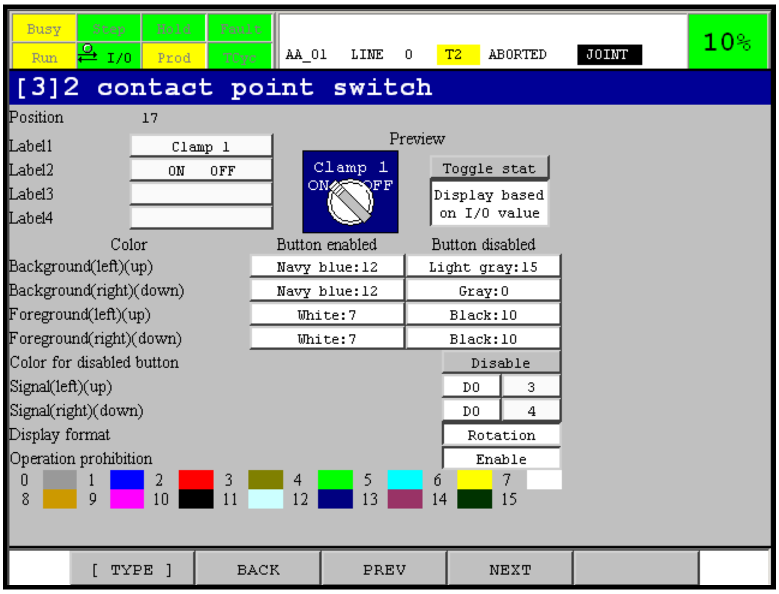
If both signals are ON or OFF, you will see button similar to the following figure.



NOTE

The 2 signals setup for 2 contact point switch should not be ON (or OFF) at the same time.
It is not proper state for the type of button. If this is not expected in your system, please confirm design of your system. I/O configuration and TP program for example. If it is expected status, don’t use 2 contact point switch for the signals.

If you press the button in this status, signal (left) (up) is changed. If it is OFF, it is turned on. If it is ON, it is turned off.



[3]2 contact point switch

Position 17

Label1 Clamp 1

Label2 ON OFF

Label3

Label4

Preview

Color	Button enabled	Button disabled
Background(left)(up)	Navy blue:12	Light gray:15
Background(right)(down)	Navy blue:12	Gray:0
Foreground(left)(up)	White:7	Black:10
Foreground(right)(down)	White:7	Black:10

Color for disabled button Disable

Signal(left)(up) DO 3

Signal(right)(down) DO 4

Display format Rotation

Operation prohibition Enable

Following table is list of setup items.

Two default values are written in default value column of color items. The 1st one is value for “Button enabled” column of detail setup screen. The 2nd one is for “Button disabled” column.

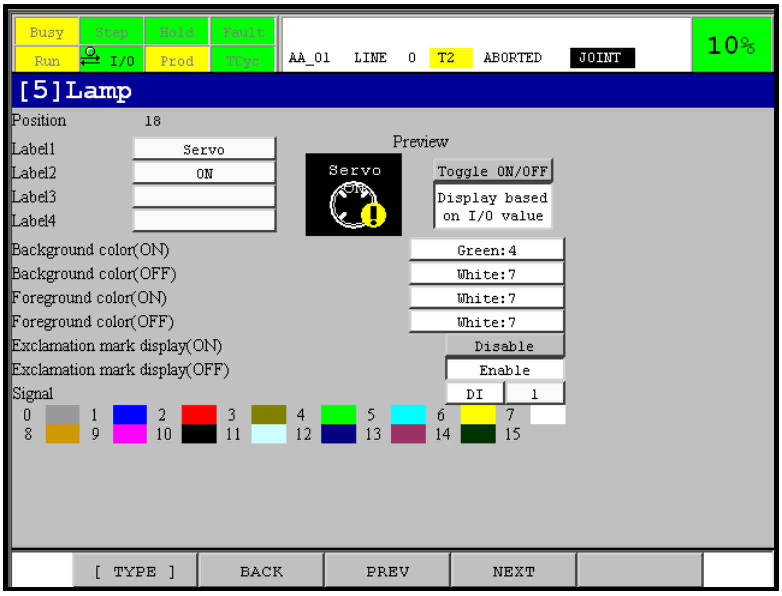
Please refer to “Foreground color and background color” of 44.8 "COMMON PROPERTIES” for more information of setup of the colors.

Item	Default value	Description
Label1, …, Label4	Label1,…,Label4	These are strings displayed on the button. If format is Up/Down, Label 4 is not used.
Background (left)(up)	Navy blue (12) and Light gray(15)	Background color of when signal (left) (up) is ON.
Background (right)(down)	Navy blue (12) and G ray(0)	Background color of when signal (right) (down) is ON.
Foreground (left)(up)	White(7) and Black(10)	Foreground color of when signal (left) (up) is ON. This is color of label.
Foreground (right)(down)	White(7) and Black(10)	Foreground color of when signal (right) (down) is ON.
Color for disabled button	Disable	When this item is enabled, color of “Button disabled” column is used when the button is disabled. Please refer to section 44.8 "COMMON PROPERTIES” for details.
Signal (left) (up)	DO[0]	Output signal to be turned on and off by this button. Please refer to the explanation at the beginning of this section.
Signal (right) (down)	DO[0]	
Display format	Rotation	The choices are “Rotation” and “Up/Down”. Please refer to the explanation at the beginning of this section.
Operation condition		“Detail” button displays operation condition screen.

If index of signal (right) (down) is 0, this button works as switch that just turns on and off signal (left) (up). Even if setup of signal (left) (up) is correct and index of signal (right) (down) is 0, it isn’t recognized as error.

44.13 LAMP

Type code is 5. This is lamp to show status of signal.



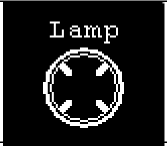
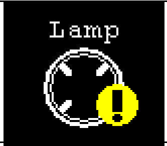
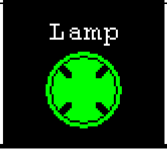
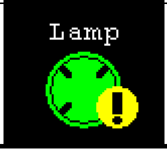
[5]Lamp

Position 18
Label1 Servo
Label2 ON
Label3
Label4
Background color(ON) Green:4
Background color(OFF) White:7
Foreground color(ON) White:7
Foreground color(OFF) White:7
Exclamation mark display(ON) Disable
Exclamation mark display(OFF) Enable
Signal DI 1

Item	Default value	Description
Label1, …, Label4	Label1,…,Label4	These are strings displayed on the lamp.
Background (ON)	Green(4)	Background color of when specified signal is ON. This is color of circular part at center of the lamp.
Background (OFF)	White(7)	Background color of when specified signal is OFF.
Foreground (ON)	White(7)	Foreground color of when specified signal is ON. This is color of label.
Foreground (OFF)	White(7)	Foreground color of when specified signal is OFF.
Exclamation mark display (ON)	Disable	Please refer to the explanation just after this table.
Exclamation mark display (OFF)	Enable	Please refer to the explanation just after this table.
Signal	DI[0]	This lamp shows status of this signal.

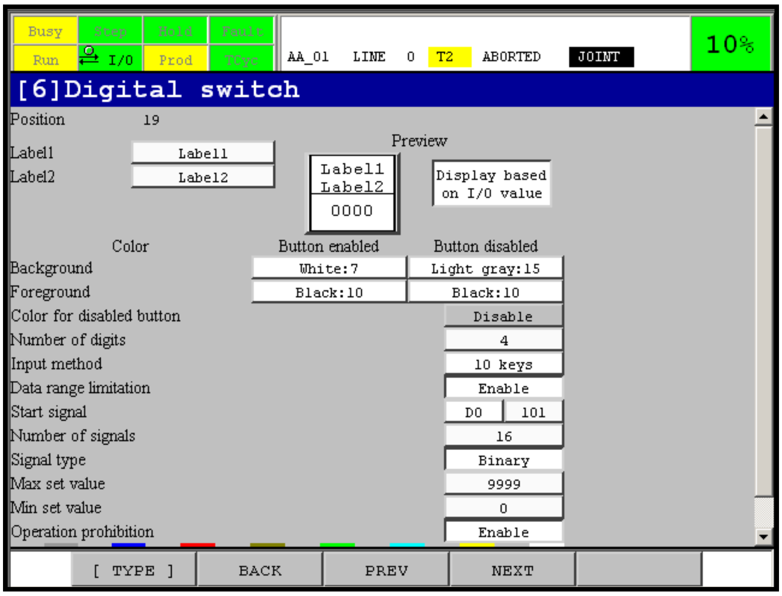
Exclamation mark display(ON) and (OFF)

These items specify if exclamation mark is displayed or not when signal is ON or OFF, respectively.

Signal	Exclamation mark display is disabled	Exclamation mark display is enabled
OFF		
ON		

44.14 DIGITAL SWITCH

Type code is 6. This switch regards a series of output signals as one group. This switch output the signals as number of specified format.



[6]Digital switch

Position 19

Label1 Label1

Label2 Label2

Preview

Label1
Label2
0000

Display based on I/O value

Color	Button enabled	Button disabled
Background	White:7	Light gray:15
Foreground	Black:10	Black:10
Color for disabled button		Disable
Number of digits		4
Input method		10 keys
Data range limitation		Enable
Start signal		DO 101
Number of signals		16
Signal type		Binary
Max set value		9999
Min set value		0
Operation prohibition		Enable

Following table is list of setup items.
Two default values are written in default value column of color items. The 1st one is value for “Button enabled” column of detail setup screen. The 2nd one is for “Button disabled” column.
Please refer to “Foreground color and background color” of Section 44.8 "COMMON PROPERTIES” for more information of setup of the colors.

Item	Default value	Description
Label1, Label2	Label1, Label2	These are strings displayed on the button.
Background	White(7) and Light gray(15)	Background color of this button.
Foreground	Both Black(10)	Foreground color of this button.
Color for disabled button	Disable	When this item is enabled, color of “Button disabled” column is used when the button is disabled. Please refer to section 44.8 "COMMON PROPERTIES” for details.
Number of digits	4	Value is displayed in specified number of digits.
Input method	10 keys	Choices are “10 keys” and “Switch”. 10 keys When you press the button, numeric keyboard (dialog box) is displayed. Input number to set to a series of signals. Switch Press of the button increments value by one. If value is maximum value, value is not incremented.”IFPN-011 Out of range(%d - %d)” is posted.
Data range limitation	Enable	If this is enabled, value is limited between maximum and minimum value, which are setup in this screen. Even if this is disabled, you cannot set value that cannot be represented by specified number of signals.
Start signal	DO [0]	This button treats a series of signals as a group to compose integer. This is the first signal of the group.

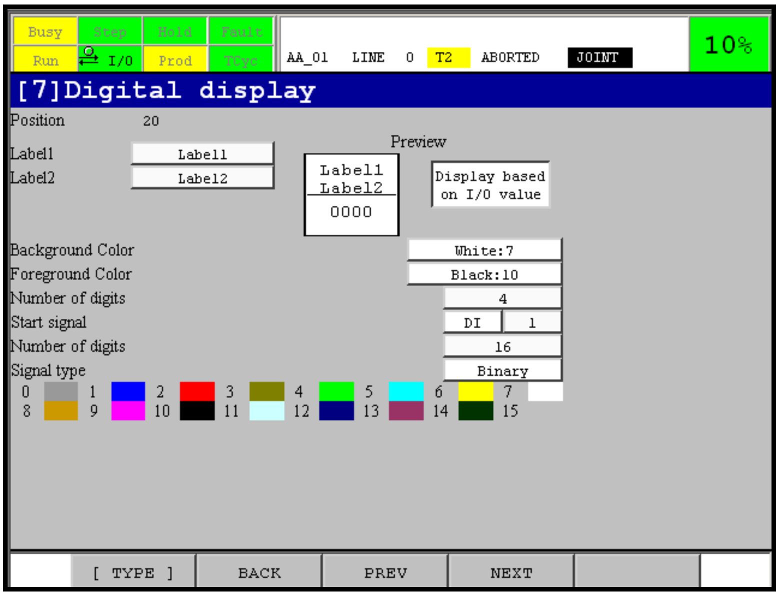
Item	Default value	Description
Number of signals	16	This is how many signals are included to group of signals. The number is counted from the signal specified by another setup item “Start signal”. If start signal is DO [1] and this item is 4, This button changes from DO [1] to DO [4].
Signal type	Binary	Encoding of signals as integer number. BCD The group of signals is considered as BCD (binary coded decimal). Binary Each signal is considered as bit of binary number. This is similar to GO [ ].
Max set value	9999	This is the maximum value that this button can set when data range limitation is enabled.
Min set value	0	This is the minimum value that this button can set when data range limitation is enabled.
Operation condition		“Detail” button displays operation condition screen.

When “Display based on I/O value” button is OFF, value in preview button as follows.

- If index of start signal is 0, *** is displayed.
- Otherwise, 0 is displayed.

44.15 DIGITAL DISPLAY

Type code is 7. This lamp regards a series of output signals as numeric value and displays the value.



[7]Digital display

Position 20

Label1 Label1

Label2 Label2

Preview

Label1
Label2
0000

Display based on I/O value

Background Color White:7

Foreground Color Black:10

Number of digits 4

Start signal DI 1

Number of digits 16

Signal type Binary

0 1 2 3 4 5 6 7
8 9 10 11 12 13 14 15

Following table is list of setup items.

Item	Default value	Description
Label1, Label2	Label1, Label2	These are strings displayed on the lamp.
Background	White(7)	Background color of this lamp.
Foreground	Black (10)	Foreground color of this lamp.
Number of digits	4	Value is displayed in specified number of digits.
Start signal	DI [0]	This lamp treats a series of signals as a group. This is the first signal of the group.

Item	Default value	Description
Number of signals	16	This is how many signals are included to group of signals. Please refer to explanation of the same item of digital switch (section 44.14).
Signal type	Binary	This is encoding of signals. BCD The group of signals is considered as BCD (binary coded decimal). Binary Each signal is considered as bit of binary number. This is similar to GI [ ]. Ver. The number of “Binary” is divided by 100 and is displayed as real number to the second decimal place. The number of digits must be 3 or 4.

When “Display based on I/O value” button is OFF, value in preview button as follows.

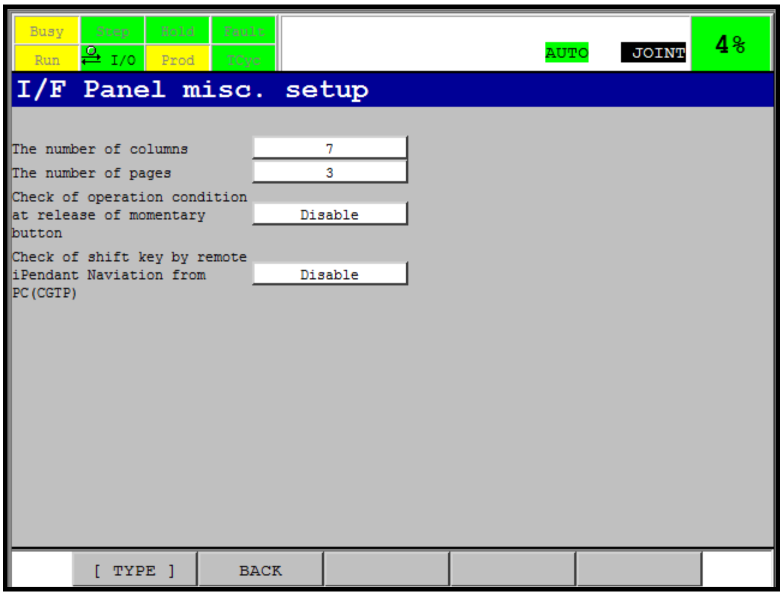
- If index of start signal is 0, *** is displayed.
- Otherwise, 0 is displayed.

44.16 MISCELLANEOUS SETTING SCREEN

Miscellaneous setting screen is a screen to setup items that affects all panels. The screen is displayed by following procedure.

1 Display interface panel setup screen. Please refer to Section 44.3.

2 Press “Miscellaneous setting” button. The following screen will be displayed.



I/F Panel misc. setup

The number of columns 7

The number of pages 3

Check of operation condition at release of momentary button Disable

Check of shift key by remote iPendant Naviation from PC(CGTP) Disable

Item	Default value	Description
The number of columns	7	Press of the button displays submenu to select the number of columns. Choices are 7 or 8.
The number of pages	3	Minimum value is 1. Maximum value is 10. You can display from page 1 to the page specified here. For example, if this item is set to 3, you can display from page 1 to 3.

Item	Default value	Description
Check of operation condition at release of momentary button	Disable	Disable At release of momentary button, operation condition is not checked. For example, panel enable switch and [SHIFT] key are not checked. Enable At release of momentary button, operation condition is checked. If button is released when operation condition isn't satisfied, output signal does not change. Button becomes released status.
Check of [SHIFT] key by remote iPendant Navigation from PC (CGTP)	Disable	Disable Suppose Internet Explorer of personal computer displays Interface panel using "Navigate iPendant (CGTP)" of robot homepage. On CGTP, operation condition does not check [SHIFT] key. You can press buttons without pressing [SHIFT] key. You can also press panel enable switch without pressing [SHIFT] key. If "Color for disabled button" is enabled, change of color also ignores status of [SHIFT] key. Enable CGTP checks [SHIFT] key. You cannot press button and panel enable switch without pressing [SHIFT] key. However, you cannot press [SHIFT] key on CGTP. You can hardly operate Interface panel.

NOTE

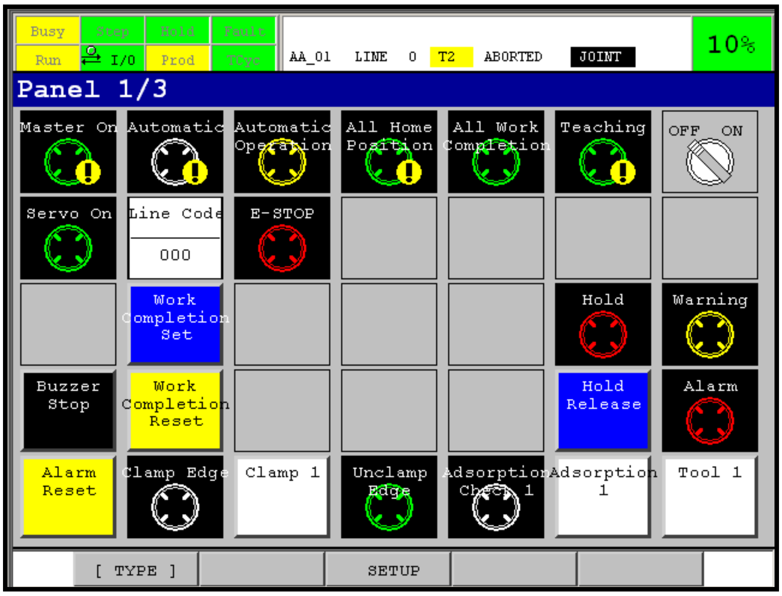
1 8 column mode can display more buttons than 7 column mode can. However, width of button is narrower.

2 Change of the number of columns causes the change of layout of buttons. Please refer to explanation in this section.

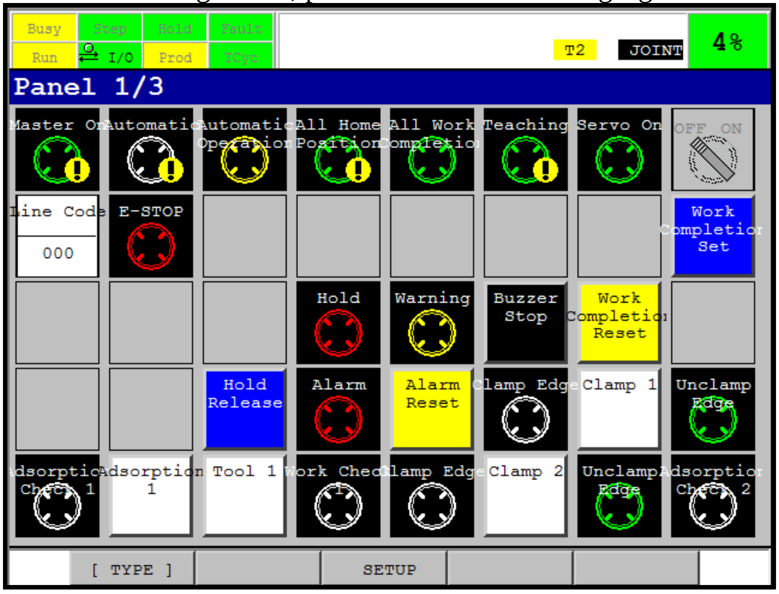
3 Don't change the number of columns and pages when the other panes are displaying Interface panel or its setup screens.

The number of columns

The number of columns of interface panel is 7 by default. You can change it to 8. Following figure is a panel of 7columns



After the number of columns is changed to 8, panel looks like following figure.



Setup of each buttons is common to both display. Both panels display buttons sequentially from position number 1. The number of buttons displayed on each panel is different.

For example, the 1st page of 7 column panel displays from position number 1 to 34. That of 8 column displays from position number 1 to 39.

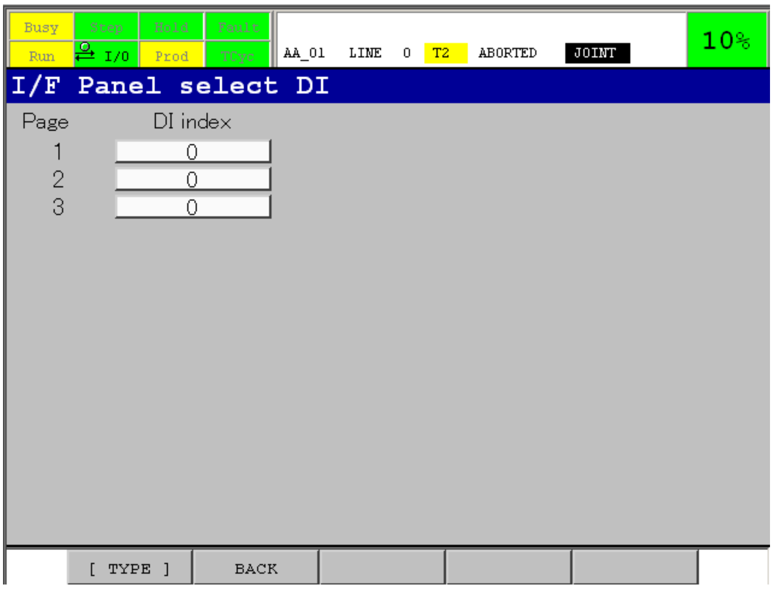
When the number of columns is 8, the last 5 buttons of the 1st page are buttons that were the first 5 buttons on the 2nd page of 7 column panel.

44.17 EXTERNAL I/F PANEL SELECTION SETUP SCREEN

You can display Interface panel at rise of digital input signal (DI). The signal is specified in external Interface panel selection setup screen. The screen is displayed by the following procedure.

1 Display interface panel setup screen. Please refer to Section 44.3.

2 Press “External I/F panel selection” button. The following screen will be displayed.



I/F Panel select DI

Page	DI index
1	0
2	0
3	0

- The number of signals alters based on the number of pages.
- At the rise of specified DI, corresponding page of Interface panel is displayed.
The panel is displayed in the 1st pane of iPendant.

- Signal is specified for each page
- Interface panel may not be displayed when this function conflicts with another function to force display of a screen.

> **NOTE**
> Cycle power is needed for change to take effect.

44.18 BACKUP AND RESTORE

Files for backup and restore are classified into two groups.

- IFP0**.XML (** is 2 characters that are language suffix).
  For example, it is “eg” for English and “kn” for Japanese.
- IFP_VARS.SV

IFP0**.XML

- This file contains setup of panel.
  Position number, type and parameters for buttons are saved into these files.
- Contents of panels are saved into file for each language.
- IFP**.XML can be restored at cold start.

> **NOTE**
> 1 Don’t load the XML file when panel or setup screens are displayed.
> 2 Setup of type setup screen and button detail setup screens are saved at exit from the screens. Backup after exiting from the screens.

IFP_VARS.SV contains following setup.

- Signal number of external Interface panel selection
- Items of miscellaneous setting screen.
- IFP_VARS.SV should be read in controlled start mode.

44.19 LIMITATIONS

1 iPendant with touch panel is required.
2 Buttons and lamp of this function only support digital type signal.
Digital type signal means signals that have ON/OFF value, DI/DO and RI/RO for example.
This function does not support I/O type that has integer value, GI/GO and AI/AO for example.
The other data of robot such as register and system variable cannot be handled by buttons of this function.
3 You cannot use “<” and “>” in string of label.
4 After pressing a button, release it after confirming change of color and pressed/released status of button.
5 After pressing a button, release touch panel on the same button.
If you drag your finger and release touch panel on another button or empty place, button doesn’t work properly.
Especially, momentary button doesn’t turn off specified signal.
Press and release the button again, properly.
6 Display panel and all setup screens in single pane mode.
Otherwise, whole screen cannot be displayed. Cursor may be displayed outside of screen.
Changed setup items that is not saved may be lost by display or operation on another pane.
You cannot change focus by touching panel and type setup screen. Use DISP key.
7 Setup of type setup screen and button detail setup screens are saved at exit from screen.
Exit from the screens before display panel, power off or backup.
8 Zoom function does not zoom text and image of button.

45 SPECIAL JOG SEQUENCE

This function enables Jog operation without pressing [SHIFT] key.
Pressing only Jog keys, the robot moves in low speed. So you can confirm the direction the robot moves in under safe condition.
Normal Jog operation is also possible by pressing [SHIFT] key additionally.
It is possible to order this function independently.
In that case, this function is available in all tools.

45.1 USAGE

When this function is valid, Jog operation is possible in the following conditions.

Press only Jog keys:

The robot moves in the speed corresponds to override 5%. When the override is less than 5%, the robot moves in the speed corresponds to the override.

Press both Jog keys and [SHIFT] key:

The robot moves in the normal Jog speed. The Jog speed corresponds to the override specified in Teach Pendant.

When this function is invalid, Jog operation is possible in the following conditions.

Press only Jog keys:

The robot does not move and the following message is displayed.

Press both Jog keys and [SHIFT] key:

The robot moves in the normal speed. The Jog speed corresponds to the override specified by Teach Pendant.

45.2 SETUP

This function can be set up by the following system variable.

$JOG_SP_MODE:

0: This function is invalid.

1-100: This function is valid. When only Jog key is pressed, the robot moves in the speed corresponds to the override specified here.

⚠**WARNING**

If large value is specified in this system variable, the robot possibly moves in high speed. For secure use, the value of this system variable should be less than 5.

45.3 APPLICATION

Automatic Override setup when changing frames:

Normally, override is automatically initialized to 10% when Jog frame is changed in Teach Pendant. With this function, as you can confirm the direction in low speed, you may want to make this automatic override setup invalid.
Setting the following system variable, the current override can be kept when frame is changed.

$SCR.$COORDOVRD = 100 (Standard value 10)

When the current override is larger than the value specified here, the override is changed into the value specified here.

46 MOTION INSTRUCTION ENHANCED EDITING

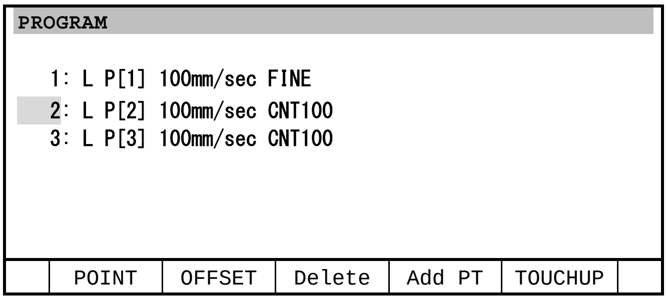
This function includes two functions, “Motion instruction insert and instruction delete function” and “Data offset function”. These are described below.
It is possible to order this function independently.
In that case, this function is available only in handling tool.

46.1 MOTION INSTRUCTION INSERT AND INSTRUCTION DELETE FUNCTION

This function enables you to insert motion instructions or delete instructions in a program with function keys in the edit screen.

46.1.1 Usage

Function keys are displayed as follows in EDIT screen.

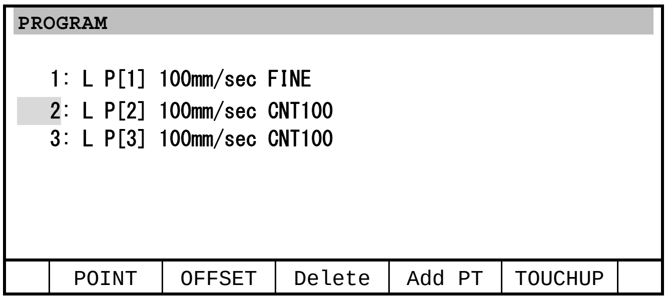


```
PROGRAM

   1: L P[1] 100mm/sec FINE
   2: L P[2] 100mm/sec CNT100
   3: L P[3] 100mm/sec CNT100
```

Procedure 46-1 Delete instructions

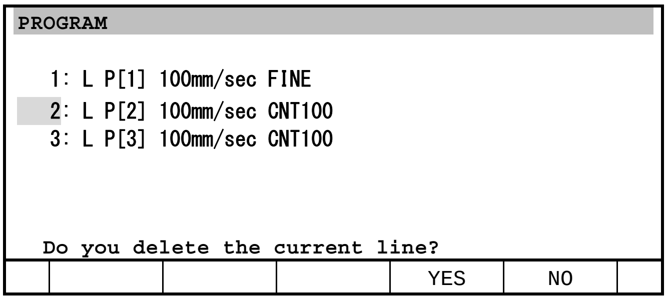
Step

1. Move the cursor on the line you want to delete.
2. Enable Teach Pendant, and press SHIFT and F3, Delete.



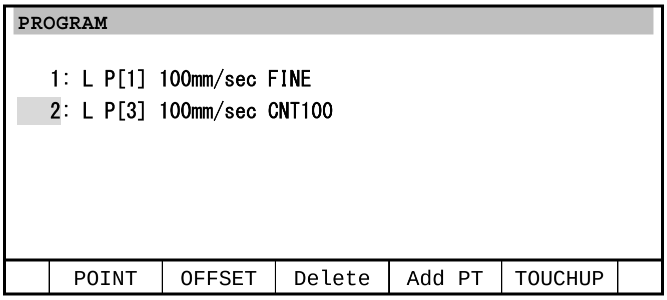
```
PROGRAM

   1: L P[1] 100mm/sec FINE
   2: L P[2] 100mm/sec CNT100
   3: L P[3] 100mm/sec CNT100
```

3. F4, YES and F5, NO will be displayed. Press YES to delete. When NO is pressed, the deleted is cancelled and edit screen will be displayed .



```
PROGRAM

   1: L P[1] 100mm/sec FINE
   2: L P[2] 100mm/sec CNT100
   3: L P[3] 100mm/sec CNT100



  Do you delete the current line?
```

4. The line that the cursor was on is deleted.

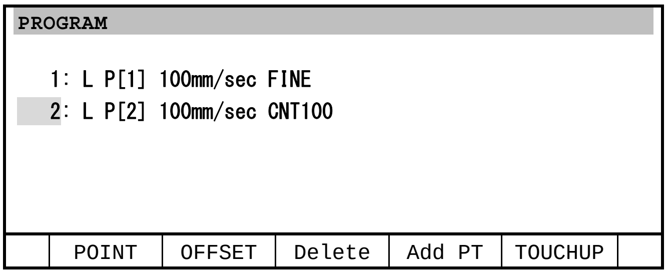


```
PROGRAM

   1: L P[1] 100mm/sec FINE
   2: L P[3] 100mm/sec CNT100
```

Procedure 46-2 Insert motion instructions

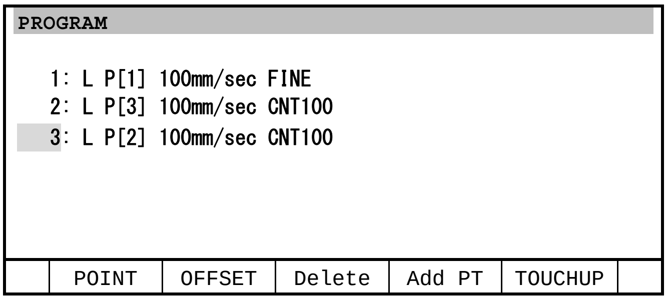
Step

1. Move the cursor on the line you want to delete.
2. Enable Teach Pendant, and press SHIFT and F4, Add PT.



```
PROGRAM

   1: L P[1] 100mm/sec FINE
   2: L P[2] 100mm/sec CNT100
```

3. Motion line is inserted, and the cursor moves to the next line.



```
PROGRAM

   1: L P[1] 100mm/sec FINE
   2: L P[3] 100mm/sec CNT100
   3: L P[2] 100mm/sec CNT100
```

46.1.2 Precautions

- Standard motion instruction that is selected in F1, POINT menu is inserted. This is the same motion instruction that is taught by pressing [SHIFT] key and POINT.
- When Circular motion is selected as standard motion instruction, Circular motion is inserted. In this case, current position is recorded in passing point and the target point is unspecified.
- After operations with this function, you cannot undo the operation by Undo in [EDCMD].

- This function cannot be used in quick menu.

46.2 DATA OFFSET FUNCTION

Data offset function enables you to change multiple position data or motion speed at once in a program that you are editing. This function includes 3 functions, “Feed rate conversion”, “Position data conversion (direct method)”, and “Position data conversion (2 point teach method)”.

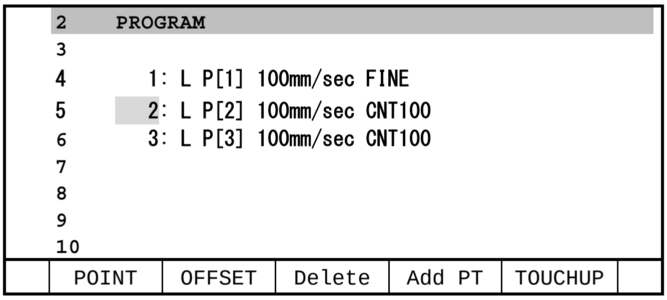
46.2.1 Feed Rate Conversion

- This function converts the feed rate of multiple motion instructions at once.
- You can convert a whole or part of the program you are editing.
- You can specify the offset to current feed rate.
- You can convert the feed rate of all motion instructions into specified value regardless of current feed rate.
- You can convert the motion instructions whose current feed rate is equal to specified value into the value you want to give.
- You can specify offset rate (%) to current feed rate.
- Except offset rate specification, only one motion type of Joint, Linear or Circular motion can be converted at once.

Procedure 46-3 Feed rate conversion

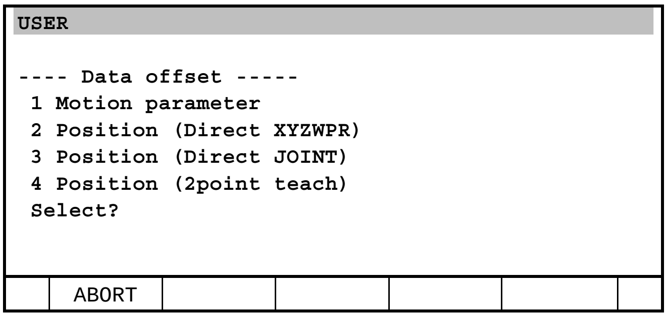
Step

1 Press F2, OFFSET in EDIT screen.



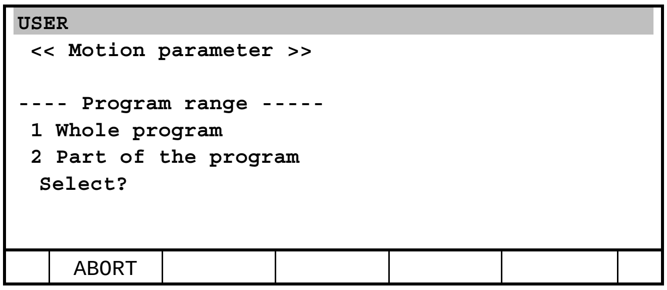
```
2    PROGRAM
3
4       1: L P[1] 100mm/sec FINE
5       2: L P[2] 100mm/sec CNT100
6       3: L P[3] 100mm/sec CNT100
7
8
9
10
```

2 Input “1” to select “1 Motion parameter”.



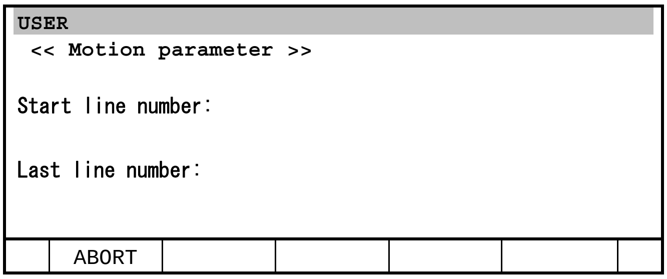
```
USER

---- Data offset -----
 1 Motion parameter
 2 Position (Direct XYZWPR)
 3 Position (Direct JOINT)
 4 Position (2point teach)
 Select?
```

3 To convert a whole program, input “1”. Then go to step 5. To convert a part of a program, input “2”. Then go to step 4.



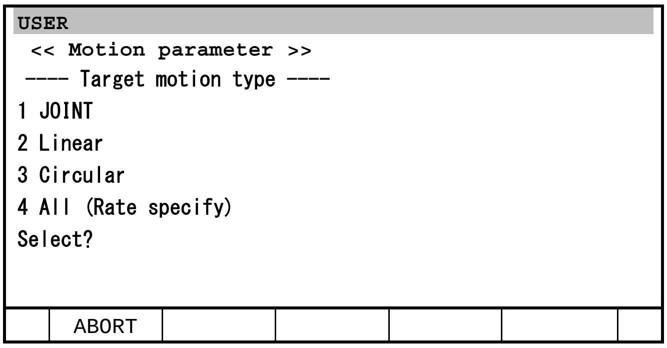
```
USER
 << Motion parameter >>

---- Program range -----
 1 Whole program
 2 Part of the program
  Select?
```

4 Input start line number and last line number.



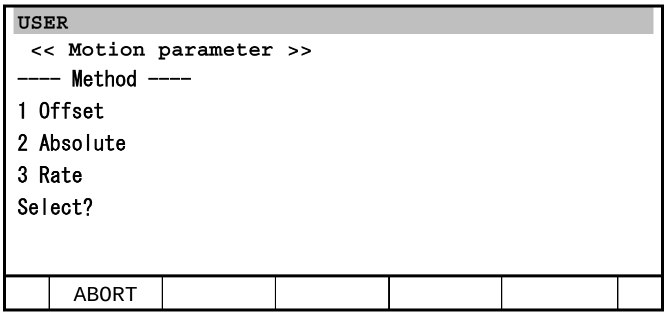
```
USER
 << Motion parameter >>

Start line number:

Last line number:
```

5 Select target motion type or All (Rate specify). If you select All (Rate specify), go to step 11.



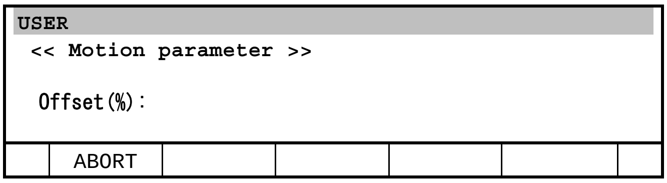
```
USER
 << Motion parameter >>
 ---- Target motion type ----
1 JOINT
2 Linear
3 Circular
4 All (Rate specify)
Select?
```

6 Select the method for converting feed rate.
Input “1” to specify the offset to the current feed rate. Then go to step 7.
Input “2” to convert feed rate to specified value. Then go to step 8.
Input “3” to specify the rate to the current feed rate. Then go to step 11.



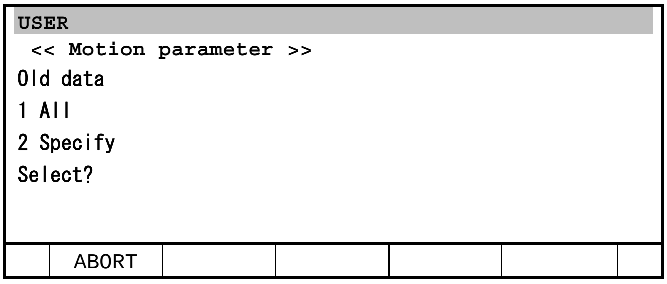
```
USER
 << Motion parameter >>
---- Method ----
1 Offset
2 Absolute
3 Rate
Select?
```

7 Input the offset. When you want to decrease the value, input negative value. The unit is % when Joint motion or mm/sec when Linear motion or Circular motion. Go to step 12.



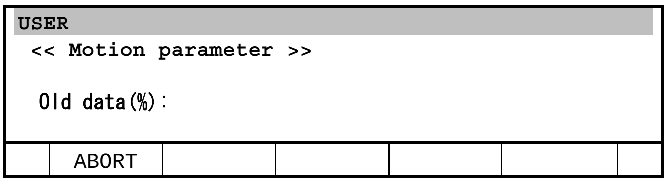
```
USER
 << Motion parameter >>

  Offset(%):
```

8 Input “1” to convert all the motion instructions. Then go to step 10.Input “2” to convert only the motion instructions whose current feed rate is equal to the value specified by operation 9. Then go to step 9.



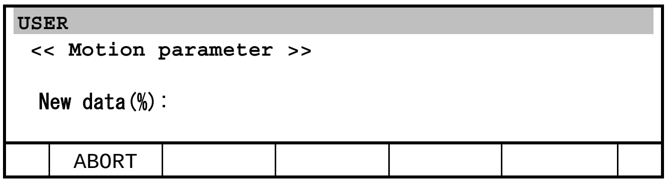
```
USER
 << Motion parameter >>
Old data
1 All
2 Specify
Select?
```

9 Input current feed rate of the motion instructions that you convert. The unit is % when Joint motion or mm/sec when Linear or Circular motion.



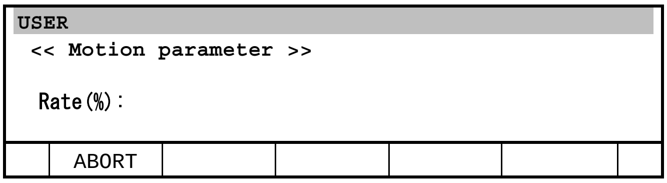
```
USER
 << Motion parameter >>

  Old data(%):
```

10 Input new value of feed rate. The unit is % when Joint motion or mm/sec when Linear or Circular motion. Go to step 12.



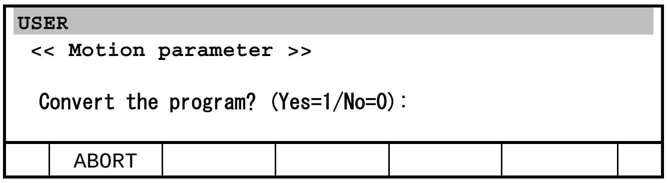
```
USER
 << Motion parameter >>

  New data(%):
```

11 Specify the rate (-100 to 100).



```
USER
 << Motion parameter >>

  Rate(%):
```

12 Input “1” to convert the program. Input “0” to cancel the conversion.



```
USER
 << Motion parameter >>

  Convert the program? (Yes=1/No=0):
```

13 Edit screen will be displayed after conversion.

46.2.2 Position Data Conversion (direct method)

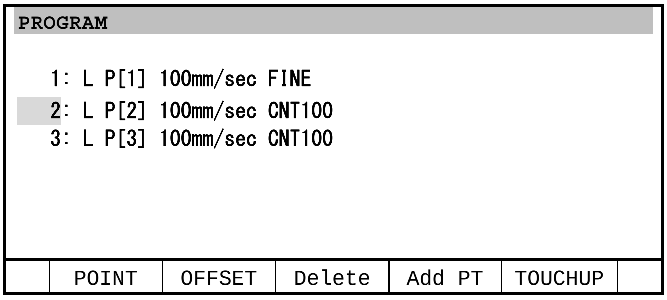
Position data conversion (2 point teach method) converts multiple position data in the program at once. Translation is specified by teaching 2 point, the original position and the destination position.

- You can convert a whole or part of the program you are editing.
- You can specify the offset to current position data.
- You can convert the position data of all motion instructions into specified value regardless of current position data.
- You can convert the motion instructions whose current position data is equal to specified value into the value you want.
- You can specify the number of user frame to specify the value.
- You can select the type, Cartesian (X, Y, Z, W, P, R) or Joint (J1 to J9).
- Only one element of X, Y, Z, W, P, R or J1 to J9 can be converted at once.

Procedure 46-4 Position data conversion (direct method)

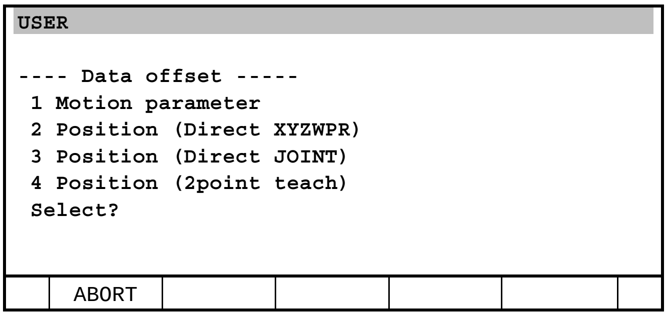
Step

1 Press F2, OFFSET in EDIT screen.



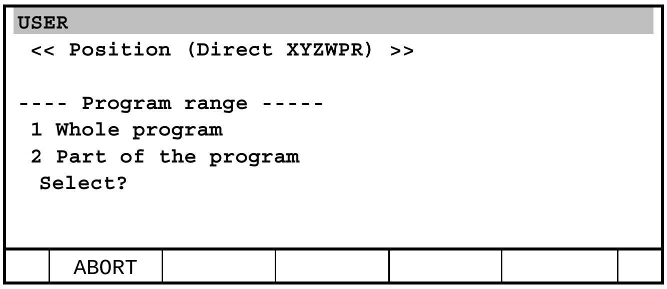
```
PROGRAM

   1: L P[1] 100mm/sec FINE
   2: L P[2] 100mm/sec CNT100
   3: L P[3] 100mm/sec CNT100
```

2 Input “1” to select “2 Position (Direct XYZWPR). Input “2” to select “3 Position (Direct JOINT).



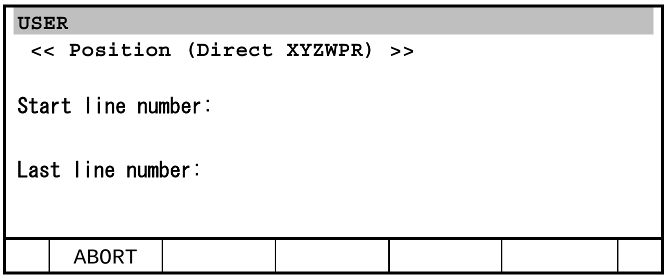
```
USER

---- Data offset -----
 1 Motion parameter
 2 Position (Direct XYZWPR)
 3 Position (Direct JOINT)
 4 Position (2point teach)
 Select?
```

3 Input “1” to convert a whole program. Then go to step 5. Input “2” to convert a part of a program. Then go to step 4.



```
USER
 << Position (Direct XYZWPR) >>

---- Program range -----
 1 Whole program
 2 Part of the program
  Select?
```

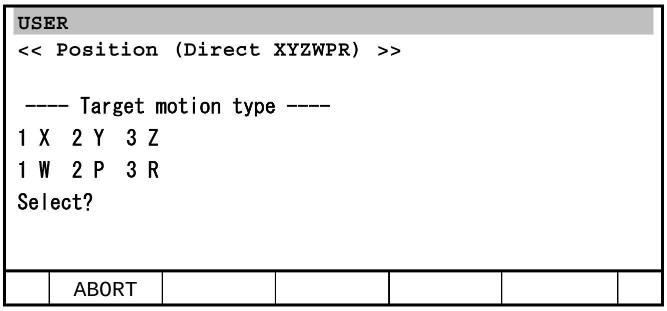
4 Input start line number and last line number.



```
USER
 << Position (Direct XYZWPR) >>

Start line number:

Last line number:
```

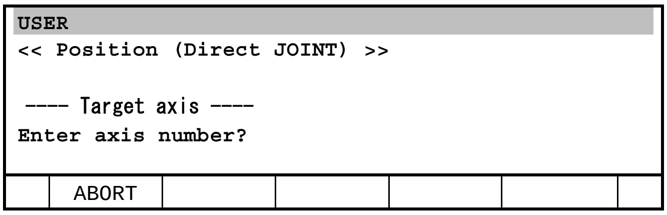
5 Input the number of the element or axis number to convert.

When “Direct XYZWPR” is selected



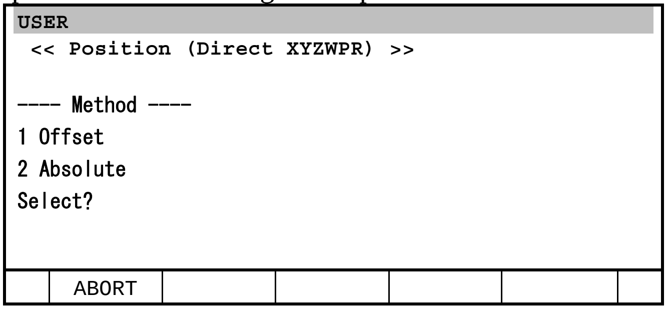
```
USER
<< Position (Direct XYZWPR) >>

 ---- Target motion type ----
1 X  2 Y  3 Z
1 W  2 P  3 R
Select?
```

When “Direct JOINT” is selected



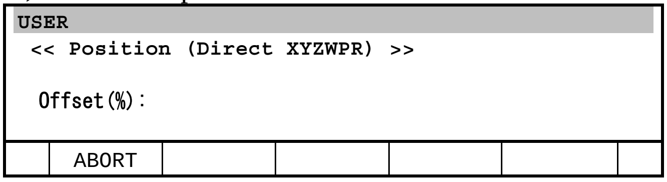
```
USER
<< Position (Direct JOINT) >>

 ---- Target axis ----
Enter axis number?
```

6 Input “1” to specify the offset to current position data. Then go to step 7. Input “2” to specify the position data into specified value. Then go to step 8.



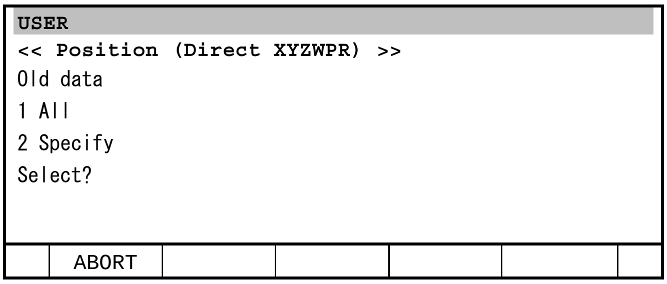
```
USER
 << Position (Direct XYZWPR) >>

---- Method ----
1 Offset
2 Absolute
Select?
```

7 Input the offset. When you want to decrease the value, input negative value. The unit is mm when X, Y, Z or deg when W, P, R. Go to step 11.



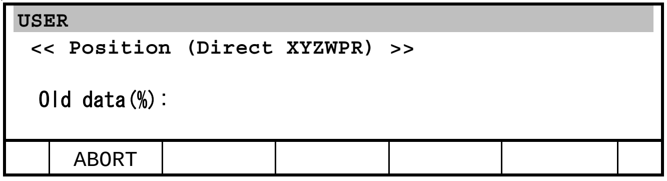
```
USER
 << Position (Direct XYZWPR) >>

  Offset(%):
```

8 Input “1” to convert all the motion instructions. Then go to step 10. Input “2” to convert only the motion instructions whose current feed rate is equal to the value specified by operation 9. Then go to step 9.



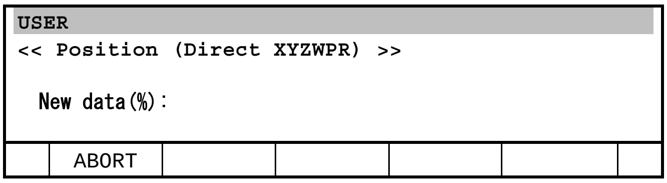
```
USER
<< Position (Direct XYZWPR) >>
Old data
1 All
2 Specify
Select?
```

9 Input the current position data of the motion instruction that you convert. The unit is mm when X, Y, Z or deg when W, P, R.



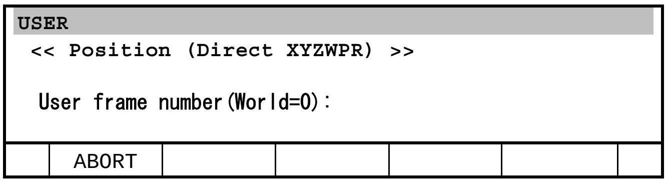
```
USER
 << Position (Direct XYZWPR) >>

  Old data(%):
```

10 Input new position data. The unit is mm when X, Y, Z or deg when W, P, R.



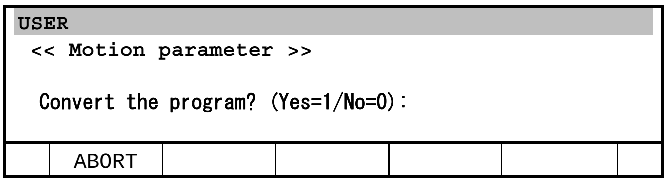
```
USER
<< Position (Direct XYZWPR) >>

  New data(%):
```

11 Input the user frame number that is the base of conversion. When world frame is used, input "0".



```
USER
 << Position (Direct XYZWPR) >>

  User frame number(World=0):
```

12 Input "1" to convert the program. Input "0" to cancel the conversion.



```
USER
 << Motion parameter >>

  Convert the program? (Yes=1/No=0):
```

13 Edit screen will be displayed after conversion.

46.2.3 Position Data Conversion (2 point teach method)

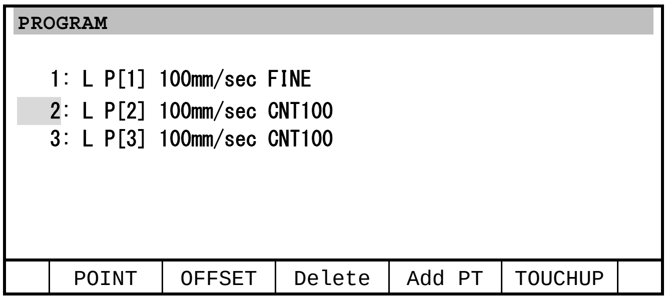
Position data conversion (2 point teach method) converts multiple position data in the program at once. Translation is specified by teaching 2 point, the original position and the destination position.

- You can convert a whole or part of the program you are editing.
- Parallel element (X, Y, Z) is converted on world frame and rotation element (W, P, R) is converted on tool frame.

Procedure 46-5 Position data conversion (2 point teach method)

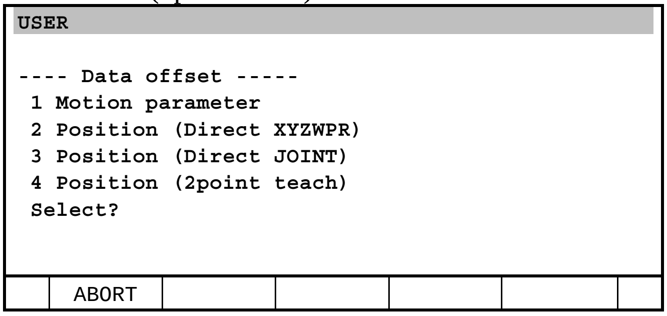
Step

1 Press F2, OFFSET in EDIT screen.



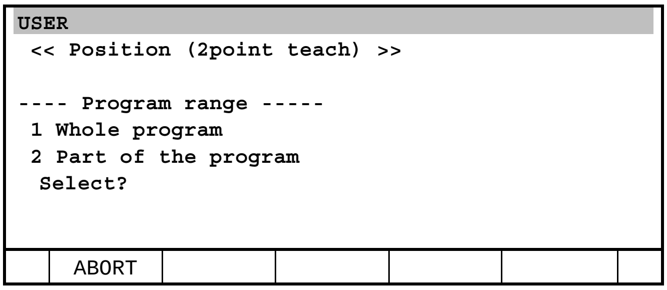
```
PROGRAM

   1: L P[1] 100mm/sec FINE
   2: L P[2] 100mm/sec CNT100
   3: L P[3] 100mm/sec CNT100
```

2 Input “4” to select “4 Position (2point teach)”.



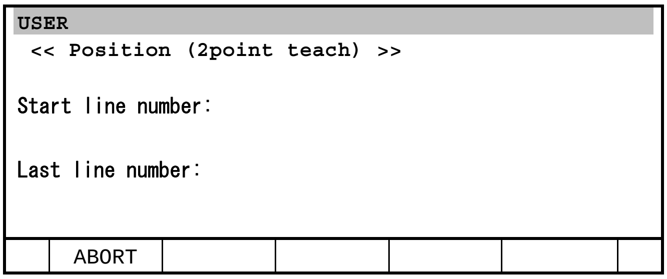
```
USER

---- Data offset -----
 1 Motion parameter
 2 Position (Direct XYZWPR)
 3 Position (Direct JOINT)
 4 Position (2point teach)
 Select?
```

3 Input “1” to convert a whole program. Then go to step 5. Input “2” to convert a part of a program. Then go to step 4.



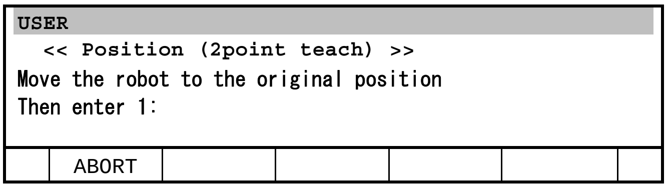
```
USER
 << Position (2point teach) >>

---- Program range -----
 1 Whole program
 2 Part of the program
  Select?
```

4 Input start line number and last line number.



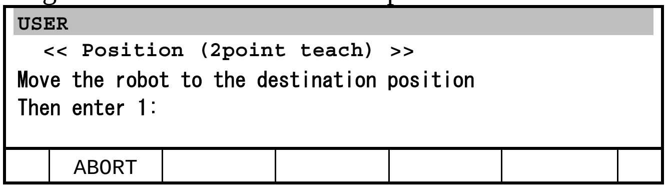
```
USER
 << Position (2point teach) >>

Start line number:

Last line number:
```

5 Input “1” after moving the robot to the original position.



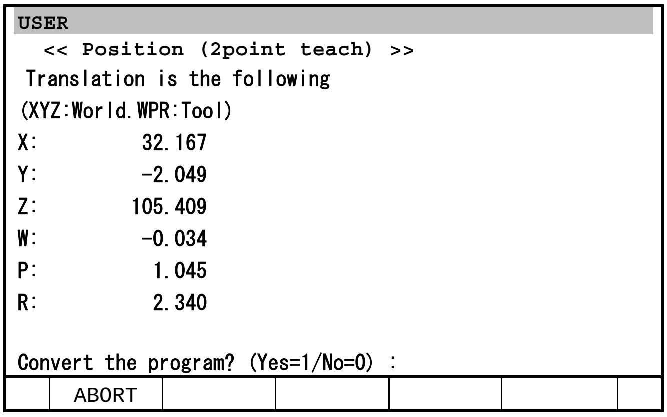
```
USER
  << Position (2point teach) >>
Move the robot to the original position
Then enter 1:
```

6 Input “1” after moving the robot to the destination position.



```
USER
  << Position (2point teach) >>
Move the robot to the destination position
Then enter 1:
```

7 Each translation is displayed.
The value X, Y, Z is the translation on world frame.
The value W, P, R is the translation on tool frame.
Input “1” to convert the program.
Input “0” to cancel the conversion.



```
USER
  << Position (2point teach) >>
 Translation is the following
(XYZ:World.WPR:Tool)
X:         32.167
Y:         -2.049
Z:        105.409
W:         -0.034
P:          1.045
R:          2.340

Convert the program? (Yes=1/No=0) :
```

8 Edit screen will be displayed after conversion.

46.2.4 Precautions

- After operations with this function, you cannot undo the operation by Undo in [EDCMD].
- The original program is copied to YFDTOFBK. When you want to use the original program, copy YFDTOFBK.
- A program whose name is YFDTOFBK is overwritten after conversion. Please do not create YFDTOFBK.
- In the multi group system, only the position data of first group can be converted.
- When the position data is converted in a system with extended axes, only the position data of robot axes are converted. The position data of extended axes are not converted.
- When another screen is displayed (ex current position display) during the operation of data offset function, you can return to the operation of data offset function by pressing EDIT key, or selecting “9 USER” in MENU.
- Press F1, ”ABORT” when you want to abort data offset function and return to EDIT screen. When you press F2, “OFFSET” again after this operation, you can restart data offset function from the beginning.
- When teach pendant is disabled, data offset function does not work by pressing F2, “OFFSET”. When teach pendant is turned disabled during the operation of data offset function, the operation is aborted and EDIT screen is displayed.
- When the position data is singular point or out of stroke limit after conversion, the position data will be unspecified data. In this case, the number of the unspecified data will be displayed and displayed as follows “Error occurred(1:OK 0:Cancel)”. When you select 1, the position data that have no error will be normally converted and the position data that have errors will be unspecified data. When you select 0, the conversion will be canceled.
- In conversion of position data by “Position (Direct XYZWPR)” or “Position (2point teach)”, the amount of the change in all axes is automatically adjusted not to exceed plus and minus 180 degree.
- When position data is converted, the position data is recorded as Cartesian coordinates.

- When position register is used in the motion instruction, this function does not convert the position data of the position register.
- This function does not convert the position data of Palletizing function.
- This function does not convert the position data of Asynchronous operation group instruction or Synchronous operation group instruction.
- This function does not convert the position data of Incremental instruction correctly. Please do not use this function to convert the position data of Incremental instruction.
- This function does not change the feed rate of the following motion instruction correctly. Please do not use this function to change the feed rate of the following motion instruction.
  - Feed rate is specified by a register.
  - Feed rate unit of Joint motion instruction is not %.
  - Feed rate unit of motion instruction except Joint is not mm/sec.

47 FINISHING FUNCTION PACKAGE

The description of option software “Finishing function package(J558)” is written in this chapter. This software is package of two optional software of Special Jog Sequence (J559) and Motion Instruction Enhanced Editing(J560).

NOTE

Finishing Function Package can be used only in handling tool.

- Please refer to Chapter 45 SPECIAL JOG SEQUENCE on this manual if you want to know about “Special Jog Sequence(J559)”.
- Please refer to Chapter 46 MOTION INSTRUCTION ENHANCED EDITING on this manual if you want to know about “Motion instruction Enhanced Editing(J560)”.

48 JOINT POSITION OUTPUT FUNCTION

Joint Position Output Function outputs current commanded position information of a specified axis with group output signal (GO).

48.1 HOW TO USE

48.1.1 Setting by System Variables

If you would like to use this function, please set following system variables on SYSTEM Variables screen.

- $SV_GUN_CTRL[n].$OUTPUT_ENB
  This is the flag to enable or disable this function. When this variable is TRUE, this function is enabled.

- $SV_GUN_CTRL[n].$GROUP_NUM
  This is used to specify the group number.

- $SV_GUN_CTRL[n].$AXIS_NUM
  This is used to specify the axis number.

- $SV_GUN_CTRL[n].$GO_VALUE
  This is used to specify the index of GO.

- $SV_GUN_CTRL[n].$IO_SCALE
  This is used to change precision of the value which stored in GO. You need to set this system variable to a positive integer. The default value of this system variable is “10”. The specified axis value is multiplied by this system variable and decimals are omitted from the result. The absolute value of the result is stored in GO.

“n” is integer from 1 to 5. It is possible to specify up to 5 axes freely with the above system variables.

48.1.2 Note

- Up to 5 axes can be specified.
- You need to assign 16 digital signals for the specified GO in advance.
- If a specified axis is linear axis, the unit is [mm]. If a specified axis is rotary axis, the unit is [deg].
- Default value of $SV_GUN_CTRL[n].$IO_SCALE is 10, so default precision is 0.1[mm] or 0.1[deg].
- The specified axis value is multiplied by this system variable and decimals are omitted from the result. The absolute value of the result is stored in GO. When the specified axis value is negative, it is defined by [32768 + (the absolute value of the result)].
- The specified axis value is multiplied by this system variable and decimals are omitted from the result. When the absolute value of the result exceeds 32767, GO is not updated.

48.1.3 Example of a Setting

In the case that store the value of axis 1 of group 2 in GO[5], the system variable is set as followings.

```
$SV_GUN_CTRL[1].$OUTPUT_ENB =   TRUE
$SV_GUN_CTRL[1].$GROUP_NUM =    2
$SV_GUN_CTRL[1].$AXIS_NUM =     1
$SV_GUN_CTRL[1].$GO_VALUE =     5
$SV_GUN_CTRL[1].$IO_SCALE =     10
```

If position of the axis is 125.64[mm], the value of GO[5] should be “1256”. If position of the axis is -80.56[mm], the value of GO[5] will be “33573 (= 32768 + 805)”.

49 EXPANDED REGISTERS FUNCTION

This option function expands the maximum number of numeric register to 5000 and position register to 2,000. However, the maximum number of position register depends on the number of group. Please refer to the table 49 for the maximum numbers of numeric and position register in each group system.

Table 49 The maximum numbers of registers in each group system

Groups	Position registers	Numeric registers
1	2000	5,000
2	1000	
3	666	
4	500	
5	400	
6	333	
7	285	
8	250	

When this option was loaded, the numbers of numeric and position register were expanded as table 49.1 automatically. And expanded numbers of registers are available in all registers instructions. Please refer to section 4.5 REGISTER INSTRUCTIONS of R-30*i*B/R-30*i*B CONTROLLER OPERATOR’S MANUAL (Basic Operation) (B-83284EN) for details of numeric and position registers function.

> **NOTE**
> This option function requires 3MB SRAM memory.

49.1 SETTING THE NUMBER OF REGISTERS

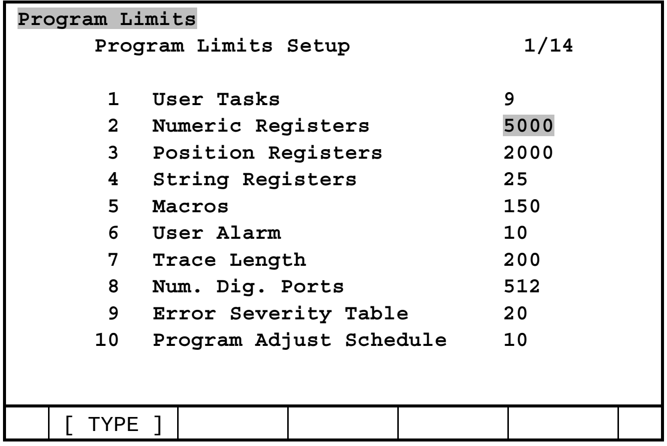
The maximum numbers of numeric and position register can be changed. However, the numbers can’t be set beyond the numbers of table 49.1.
The numbers of these registers can be change by following procedures.

Procedure 49-1 Setup the maximum numbers of numeric and position register

Step

1 Turn ON the controller with [PREV] and [NEXT] key pressed. Then select [3 Controlled start].
2 Press [MENU] key.
3 Select “0 – NEXT --” and select “1 PROGRAM SETUP”, so Program Limits screen will be displayed.

Program Limits screen



```
Program Limits
      Program Limits Setup            1/14

       1  User Tasks                9
       2  Numeric Registers         5000
       3  Position Registers        2000
       4  String Registers          25
       5  Macros                    150
       6  User Alarm                10
       7  Trace Length              200
       8  Num. Dig. Ports           512
       9  Error Severity Table      20
      10  Program Adjust Schedule   10

   [ TYPE ]
```

4 Put the cursor on numeric registers or position registers and input new value.
5 Perform a cold start.

> ⚠**WARNING**
> If the number of numeric or position register is decreased, reduced registers data are lost. For example the number of numeric registers decrease from 5,000 to 4,000, numeric registers data from 4,001 to 5,000 are lost.

49.2 SAVEING AND LOADING FILES

By executing all backup in FILE screen, numeric registers and position registers data are saved as files. When all backup is executed, numeric registers data are saved as a file named of “NUMREG.VR” and position registers data are saved named of “POSREG.VR”. By loading these files, numeric and position registers data are overwritten to the saved data. Please refer to section 8.4 SAVING FILES and 8.5 LOADING FILES of R-30*i*B/R-30*i*B CONTROLLER OPERATOR’S MANUAL (Basic Operation) (B-83284EN) for details.

However, there are some limitations to load these files. These limitations are described in following subsections.

49.2.1 LOADING NOT EXPANDED xxxREG.VR

NUMREG.VR and POSREG.VE files saved from a system not loading Expanded Registers (R830) are enabled to load to a system loading Expanded Registers (R830).

However, in this case, the numbers of numeric and position registers are translated to the numbers of loaded files.

> ⚠**WARNING**
> When you load these files, if the warning message “VARS-206 ***REG.VR will load at startup” is output, you have to reboot the controller to apply the registers data.

> **NOTE**
> At Program Limits screen in controlled start, the number of registers can be changed again. Please refer to Subsection 49.1 SETTING THE NUMBER OF REGISTERS for changing.

49.2.2 LOADING EXPANDED xxxREG.VR

Loading to the system loaded Expanded Registers

NUMREG.VR and POSREG.VE files saved from a system loading Expanded Registers (R830) are enabled to load to a system loading Expanded Registers (R830).

However, the numbers of numeric and position registers are translated to the numbers of loaded files if the number of registers is changed.

> ⚠**WARNING**
> When you load these files, if the warning message “VARS-206 ***REG.VR will load at startup” is output, you have to reboot the controller to apply the registers data.

NOTE

At Program Limits screen in controlled start, the number of registers can be changed again. Please refer to Subsection 49.1 SETTING THE NUMBER OF REGISTERS for changing.

Loading to the system NOT loaded Expanded Registers

NUMREG.VR and POSREG.VE files saved from a system loading Expanded Registers (R830) are disabled to load to a system not loading Expanded Registers (R830).

However, in the system loading Expanded Registers, if the numbers of registers are set below the values of the table 49.2, the saved file is enabled to load.

Table 49.2.2 The maximum numbers of registers in each group system

Groups	Position registers	Numeric registers
1	372	999
2	372	
3	248	
4	186	
5	148	
6	124	
7	106	
8	93	

The numbers of numeric and position registers are translated to the number of loaded file.

⚠**WARNING**

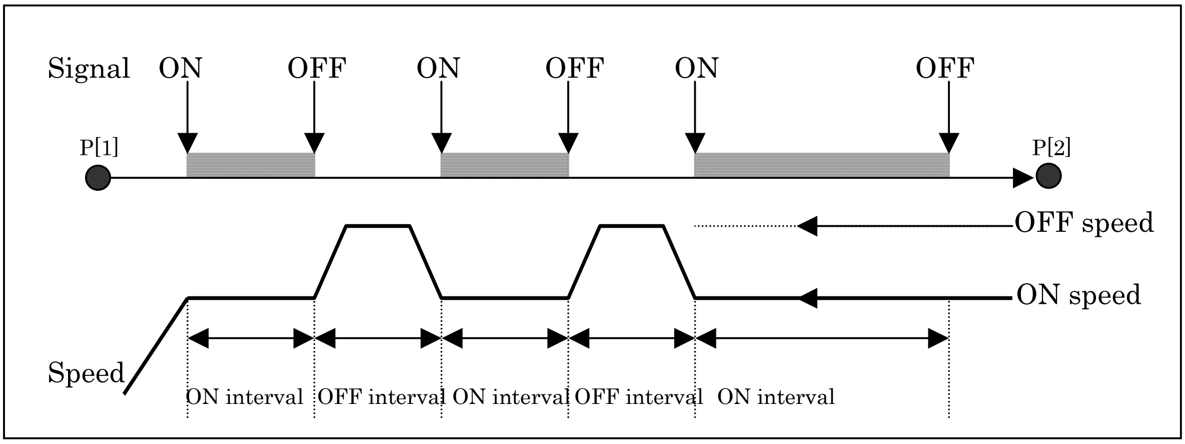
When you load these files, if the warning message “VARS-206 ***REG.VR will load at startup” is output, you have to reboot the controller to apply the registers data.

NOTE

At Program Limits screen in controlled start, the number of registers can be changed again. Please refer to Subsection 49.1 SETTING THE NUMBER OF REGISTERS for changing.

50 STITCH FUNCTION

For stitch-like sealing and laser welding, stitch function enables easy teaching of such sealing/welding part (as ON interval and OFF interval) and turning signal on/off in each interval.



50.1 SPECIFICATION

50.1.1 Instruction

Following two instructions specify an interval in which stitch function is used.
This interval is referred as stitch interval hereafter.

- **STITCH[i]**
  - Start stitch process (change of speed, signal control) according to stitch condition.
  - Stitch process doesn't stop until "STITCH END" instruction is executed.
  - Stitch condition is specified in data screen. Please refer to 50.1.2）
  - This instruction is taught as usual instruction only. This cannot be taught as motion option.
- **STITCH END**
  - By this instruction, stitch process is stopped. Signal is turned off.

Example

```
1: J P[1] 100% FINE
2:     STITCH[1]
3: L P[2] 2000mm/sec CNT100
4: L P[3] 2000mm/sec FINE
5:     STITCH END
```

Stitch interval.
Only linear and circular motions are available. If joint motion type is used, error is posted

50.1.2 Stitch Condition

- ◆ Stitch condition is set in data screen. Screens for stitch condition are Stitch Data List screen, Stitch Data Detail screen and Stitch Data Common screen.（Please refer to 50.4 APPENDIX 1 for structure of screens.）
- ◆ Items on each screen are as follows.

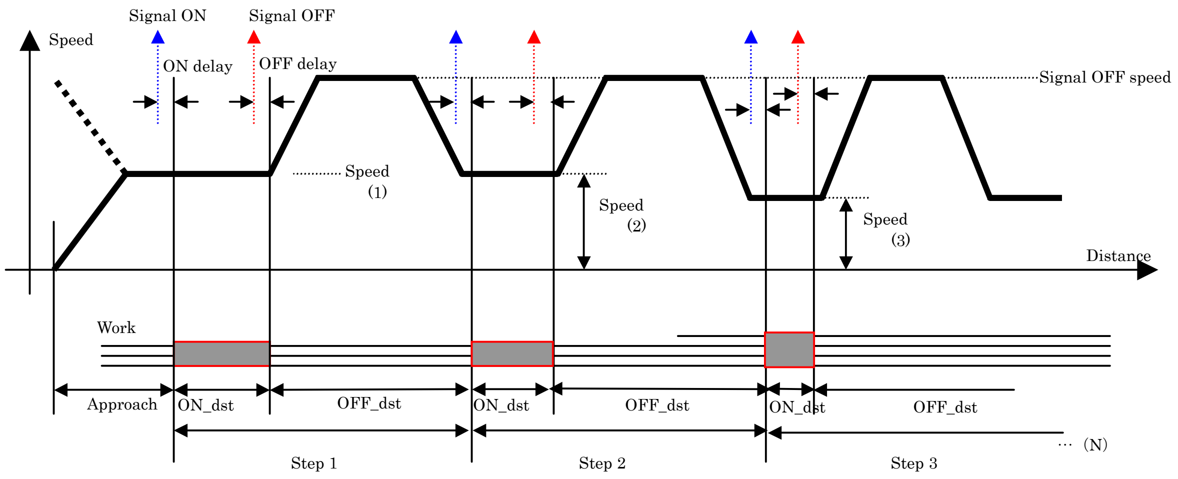
Stitch Data List screen
This screen displays list of stitch conditions. Stitch instruction specifies condition by number displayed in this screen. To display detail screen of each condition, set cursor to a condition and press F4.

Comment	Comment for each condition is displayed.	
F3（COMMON）	To display Stitch Data Common screen.	
F4（DETAIL）	To display detail screen for selected condition (condition which cursor is set to).	
F5（COPY）	User can copy selected condition to another condition. **Caution) Even if condition that selected condition is copied to is already set, the former one is overwritten.**	

Stitch Data Detail screen

◆ In this screen, following items can be set.

Item	Explanation	Range
Comment	Comment for each condition is displayed	
Equipment condition	Type of equipment signal (DO/RO/GO/AO) and index	DO/RO/GO/AO 1-32766
Signal ON delay time	Delay time between signal ON and actual action of equipment can be set here.	0-9999[msec]
Signal OFF delay time	Delay time between signal OFF and actual stop of equipment can be set here.	0-9999[msec]
Approach distance	Approach distance to stabilize speed.	0-9999[mm]
Signal OFF speed.	Speed in OFF intervals. “Speed unit” in Stitch Data Common screen decides unit.	0.1 - $PARAM_GROUP [*].$SPEEDLIM Usually, this corresponds to 2000mm/sec
Following item decide condition for each step.		
ON_dst	Distance of signal ON interval	0-9999[mm]
Speed	Speed of signal ON interval. Unit is decided in Stitch Data Common screen.	0.1 - $PARAM_GROUP[*].$SPEEDLIM Usually, this corresponds to 2000mm/sec
OFF_dst	Distance of signal OFF interval.	0-9999[mm]
Count	This specifies how many times this step is executed. If you want to repeat a sequence, input time of repetition. If 0 is specified, this step is neglected. Next step with non-zero counter is executed.	0-99[times]
Cond.	Output value of equipment signal is specified here. This value is output in ON interval.0 is output in OFF interval. If equipment signal is DO or RO, set 1. In case of GO, value specified here is output as binary code.	1-9999
F2（DATA）	You can display another Stitch Data Detail screen for different condition.	
F3（LIST）	Go back to Stitch Data List screen.	



NOTE

- Do not change ON speed and OFF speed during executing the program. Change them after pause. If it is changed during the program execution. There is the case that a robot does not work at designated speed.
- If stitch motion is performed by the controller installed TCP speed estimation function, warning “TCPP-018 Begin Error Mode at line” or “TCPP-019 Speed Ovrd Mode at line” could occur. No remedy is required for these warning.

Stitch Data Common screen

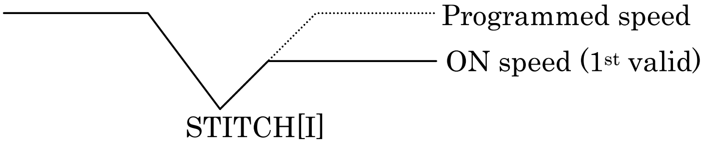
◆ In this screen, following items can be set.

Stitch enable	Only when specified signal is ON, stitch process (speed change and signal control) is performed. If this signal is OFF, speed of motion in stitch interval is programmed speed. If index is 0, it corresponds to signal ON.	DI/RI 1-32766
Equipment enable	Only when specified signal is ON, equipment signal is turned on. If this is OFF, only speed control is done. It index is 0, it corresponds to signal ON.	DI/RI 1-32766
Speed unit	Unit of Signal OFF speed and ON speed is specified here.	cm/min mm/sec inch/min

50.1.3 Flow of Stitch Process

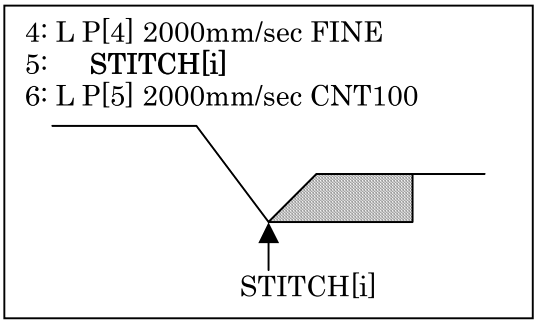
1 After execution of STITCH[i], speed of TCP is changed to ON speed of first valid step of stitch condition i. Valid step is a step with non-zero count.
Speed specified by program has no meaning.

```
4: L P[4] 2000mm/sec FINE
5:    STITCH[1]
6: L P[5] 2000mm/sec CNT100
```

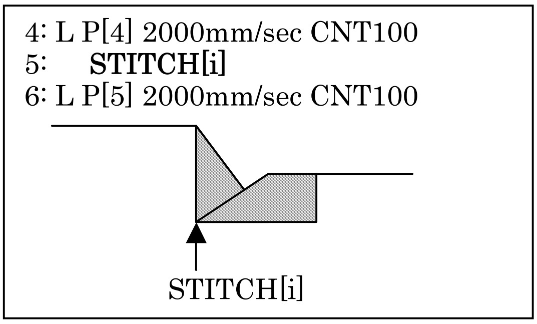


2 When distance which TCP traveled after STITCH[i] exceeds {approach distance – ON delay (converted to distance) }, R-30iA output signal.

CAUTION) If STITCH[i] instruction is executed before complete of motion, like STITCH[I] just after CNT motion, approach distance is shorter than expected.



```
4: L P[4] 2000mm/sec FINE
5:    STITCH[i]
6: L P[5] 2000mm/sec CNT100
```



```
4: L P[4] 2000mm/sec CNT100
5:    STITCH[i]
6: L P[5] 2000mm/sec CNT100
```

CAUTION) ON delay time is interpreted after <u>conversion to distance.</u> On delay time is converted to a distance that TCP travels at On speed and 100% override. Timing of signal output is based on this converted on delay time. Please adjust ON delay time with override 100%.

3 After this, change of speed and ON/OFF of signal is done according to accumulated distance TCP traveled.

4 If motion statement doesn't complete after completion of all steps, TCP moves at OFF speed until STITCH END instruction is executed.

50.1.4 Other Specifications

- Several motions in a stitch interval.

  Stitch function doesn't care how many motion statements are in a stitch interval. See following example. TCP moves from P[1] to P[3] according to stitch condition

  ```
  1: J P[1] 100% FINE
  2:    STITCH[1]
  3: L P[2] 2000mm/sec CNT100
  4: L P[3] 2000mm/sec FINE
  5:    STITCH END
  ```

- Changing stitch condition in a stitch interval

  When STITCH[i] instruction is executed in a stitch interval, New stitch condition is used from the instant. Approach distance for new stitch condition is also executed.

- Halt and resume

  When program is halted, equipment signal is turned off.

  After resume, the rest of stitch sequence is processed. But timing of stitch may be different from non-halted case due to effect of acceleration and deceleration

- Single step execution
- Backward execution
- Abort of program
- Change of line

  In these state or if one of these is done when stitch is in process, stitch function is automatically disabled. Equipment signal is turned off.

  If stitch function is disabled, programmed speed is used.

  After resume, stitch process isn't performed. To use stitch function, change execution line to one that is before STITCH[i] instruction and run halted (aborted) program.

◆ Saving stitch condition
Stitch conditions can be saved by “All of above” in file screen. File name is as follows.
SYSSTCH.SV

50.2 ADJUSTMENT

Change of speed needs time for acceleration and deceleration. For this reason, it is desired in order to maintain specified speed in ON interval:
1:to accelerate/decelerate with short acceleration/deceleration time
2:to start deceleration at appropriate timing
Speed of robot depends on specified speed. Appropriate acceleration/deceleration time depends on pose of robot even if specified speed is same.

We recommend users to adjust acceleration/deceleration time and timing of deceleration for improved performance.

1 Adjustment of acceleration and deceleration time.
Using ACC instruction, user can adjust acceleration by 1 motion statement.
It is possible to shorten acceleration/deceleration time by adding ACC instruction to motion statements in stitch interval. By ACC instruction, response of robot to change of speed is improved.

```
1: J P[1] 100% FINE
2:     STITCH[1]
3: L P[2] 2000mm/secFINE ACC150
4:     STITCH END
5: L P[3] 2000mm/sec FINE
```

CAUTION:
When Constant path function is valid ($CPCFG.$CP_ENABLE=TRUE), you can't adjust acceleration/deceleration between stitch ON and OFF using ACC instruction. If you need to adjust acceleration/deceleration, disable Constant path function ($CPCFG.$CP_ENABLE=FALSE) and cycle power of the controller. However, motion path is changed by speed override if you disable Constant path function.

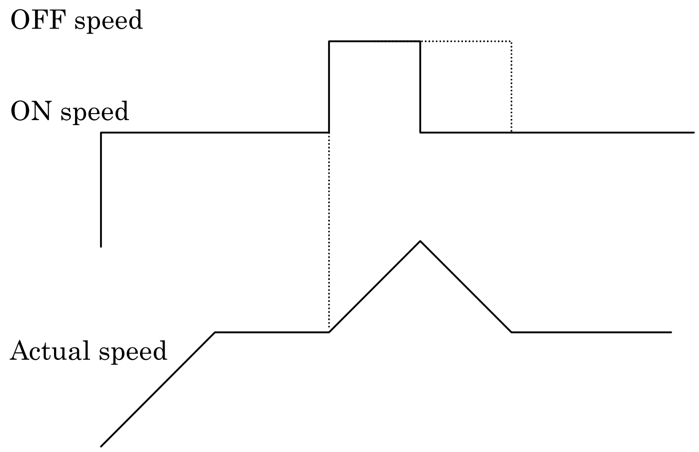
2 Timing of deceleration from OFF interval to ON interval
It is possible to decelerate earlier than usual by hastening deceleration to ON speed.
For this adjustment, following system variables are available.

$STCH[i]. $STEP[s].$DEC_NUM=1-10（default value: 2）
OFF interval is divided into pieces of this number in order to change deceleration rate by divisions.
Following system variables specify deceleration rates for each division.
$STCH[i]. $STEP[s].$DEC_RATE[1-10]（[1]=100, [2]=0, the others =100）
For each division, speed is set to following speed.
(OFF speed –On speed)*$DEC_RATE/100+On speed
This means that DEC_RATE[*] = 100 corresponds to OFF speed and 0 corresponds to ON speed.

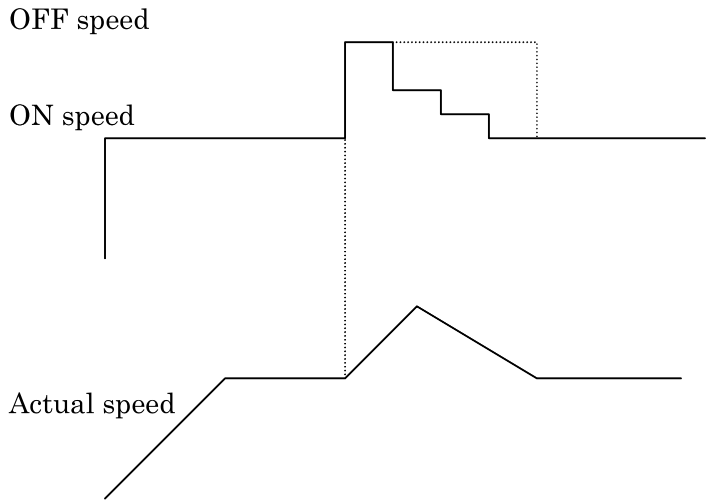
CAUTION:
DEC_RATE value should be in the range from 0 to 100.

Meaning of $DEC_NUM, $DEC_RATE[]

DEC_NUM= 2
DEC_RATE[1]=100, DEC_RATE=[2]=0



DEC_NUM= 4
DEC_RATE[1]=100, DEC_RATE=[2]=50, DEC_RATE=[3]=25, DEC_RATE=[4]=0



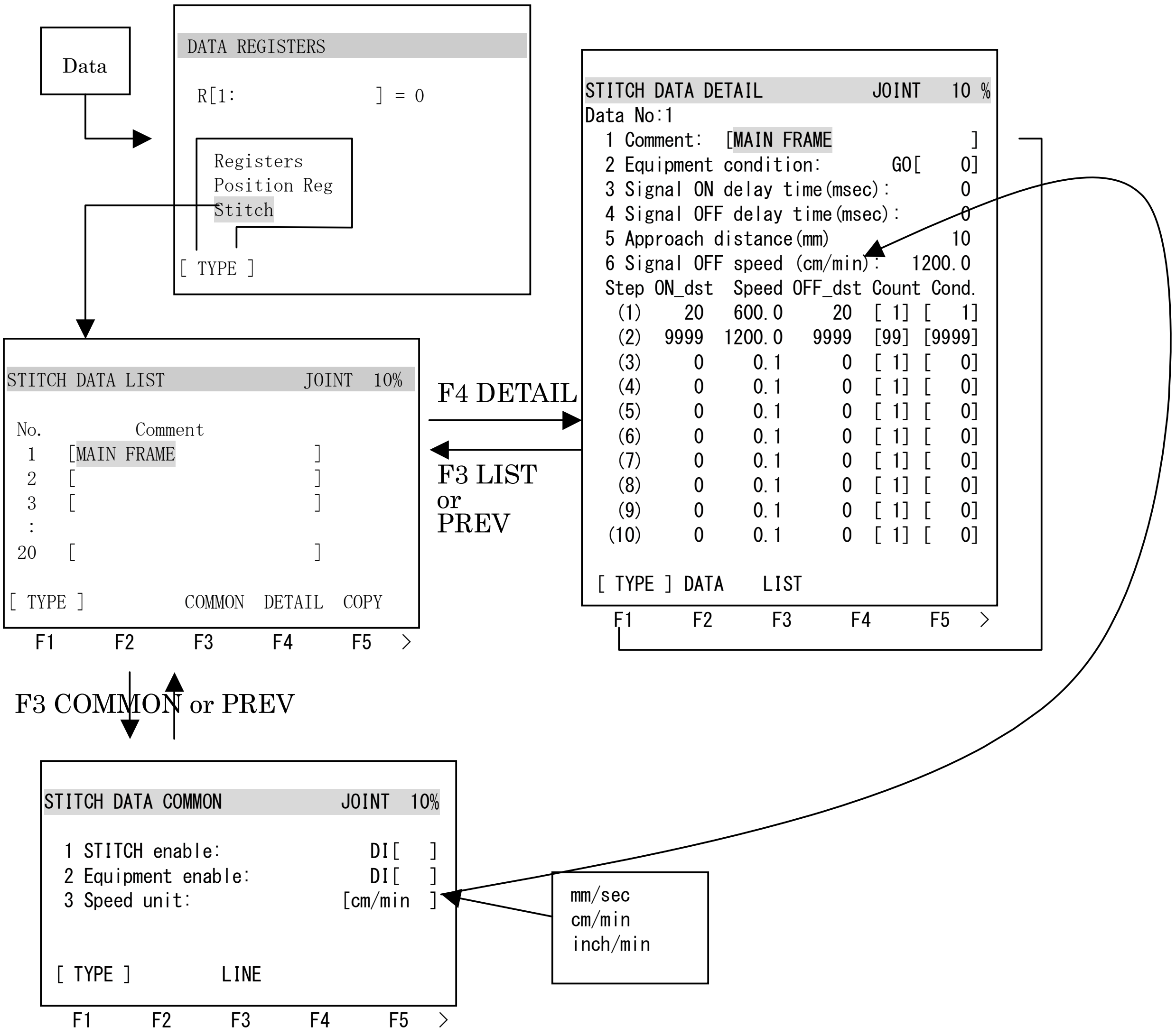
50.3 LIMITATIONS

- After E-stop, distance cannot be calculated correctly. After resume distance that TCP moved is calculated. But there may be errors due to overrun after E-stop.
- JOG doesn't accumulate distance of stitch intervals.
- In stitch interval, following operations are not available.
  JOINT motion
  Change of tool frame and user frame.
- Stitch function cannot be used with following functions.
  - ✓ Continuous turn
  - ✓ Line tracking
  - ✓ Coordinated motion
  - ✓ Remote TCP
  - ✓ Robots which doesn't have Cartesian coordination (Independent axis)
  - ✓ Extended axis
  - ✓ Distance Before (In stitch interval, Distance Before is not available.)

- ✓ Constant path function

50.4 APPENDIX

Appendix1 Structure of screens for stitch function



```
DATA REGISTERS

 R[1:                ] = 0

   Registers
   Position Reg
   Stitch

[ TYPE ]
```

```
STITCH DATA LIST                      JOINT  10%

 No.          Comment
  1   [MAIN FRAME                    ]
  2   [                              ]
  3   [                              ]
  :
 20   [                              ]

[ TYPE ]           COMMON  DETAIL  COPY
    F1       F2       F3       F4       F5    >
```

```
STITCH DATA DETAIL               JOINT   10 %
Data No:1
  1 Comment:  [MAIN FRAME                 ]
  2 Equipment condition:         GO[     0]
  3 Signal ON delay time(msec):          0
  4 Signal OFF delay time(msec):         0
  5 Approach distance(mm)               10
  6 Signal OFF speed (cm/min):     1200.0
  Step  ON_dst   Speed  OFF_dst  Count  Cond.
   (1)     20    600.0      20   [ 1] [    1]
   (2)   9999   1200.0    9999   [99] [9999]
   (3)      0      0.1       0   [ 1] [    0]
   (4)      0      0.1       0   [ 1] [    0]
   (5)      0      0.1       0   [ 1] [    0]
   (6)      0      0.1       0   [ 1] [    0]
   (7)      0      0.1       0   [ 1] [    0]
   (8)      0      0.1       0   [ 1] [    0]
   (9)      0      0.1       0   [ 1] [    0]
  (10)      0      0.1       0   [ 1] [    0]

 [ TYPE ] DATA    LIST
    F1       F2       F3       F4       F5    >
```

```
STITCH DATA COMMON                    JOINT  10%

  1 STITCH enable:                   DI[    ]
  2 Equipment enable:                DI[    ]
  3 Speed unit:                   [cm/min   ]


 [ TYPE ]         LINE
    F1       F2       F3       F4       F5    >
```

```
mm/sec
cm/min
inch/min
```

51 VISUAL DIAGNOSTICS

Visual Diagnostics uses the *i*Pendant animated picture display capability to provide a rich visual presentation of status information. A Visual Diagnostic (VD) screen is one that displays a reference image as the background, and small indicator images strategically placed and activated in the foreground. The appearance of the indicator images is driven by the state of information within the controller. This can be useful, for example, in an application in which the robot is welding. Each weld position could be monitored and visually represented so that the success or failure of a weld is reported by green and red indicators. See Fig.51 for another example.

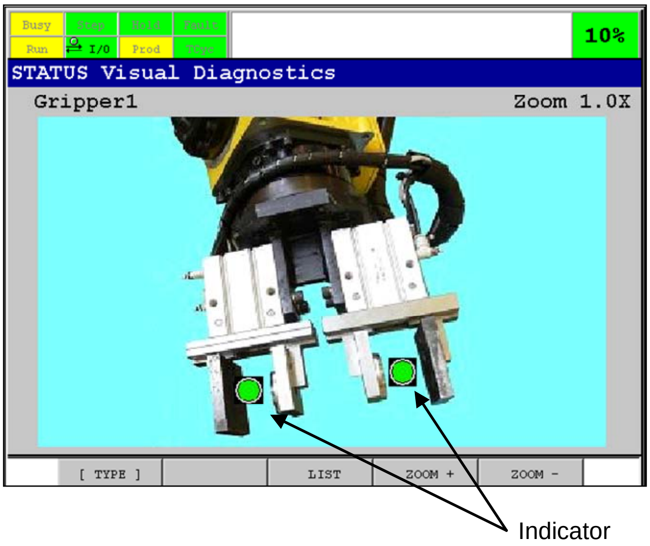


Fig.51 Simple Gripper Example

The background is a picture of a gripper with several actuators. In the foreground near many of the actuators are status indicators which are pictures of an LED to indicate the status of the actuator. The LED is green when the actuator is open and red when it is closed.

The foreground indicators are not limited to pictures of red and green LEDs. In fact, each indicator can have multiple states, each with a unique picture to display the state graphically. For example, a clamp indicator could use images of the clamp in the opened, closed and intermediate positions. If properly scaled and oriented, the clamp on the VD screen would appear to represent the configuration of the clamp accurately.

An indicator is displayed on or near an associated detail in the background image to specify its status. An indicator can have up to four states and each state can be represented by a different image. A Visual Diagnostic screen can have as many indicators as you want. However, be careful when adding indicators because each takes controller resources in terms of space and time, as well as screen clutter.

An indicator can have up to four states. An indicator's state is defined by the value of its one or two state control elements at run-time. If only one state control element (SCE) is defined (ie: number 2 is marked "**") then it can be a Boolean I/O (0=False, 1=True), Numeric I/O, Data Register or *i*Rvision Register.

When the state control element is a vision register VR[x], the model ID from view 1 is used. If, at run-time, a numeric state control element evaluates to a number other than 0 through 3, the base 4 modulus of the number is used. If both state control elements are used, then they must both be Boolean I/O types. The indicator state is then determined as shown in Table 51.

Table 51 Indicator States

SCE2	SCE1	State
0	0	0
0	1	1
1	0	2
1	1	3

Each of the four indicator states has an associated image that is displayed. If there is no state image associated with the number evaluated, the no image is shown and the background image shows through. The State Image: fields enable you to set up this state-to-image relationship.

51.1 CREATING AND EDITING VISUAL DIAGNOSTIC SCREENS

You can create custom Visual Diagnostic screens on the robot controller. Table 51.2(a) and Table 51.2(b) list items you can set when setting up a Visual Diagnostic screen. Use Procedure 51-1 to create and edit a Visual Diagnostic screen.

Procedure 51-1 Creating and Editing Visual Diagnostic Screens

Conditions

- You have graphic files accessible on the controller that you want to use as your background and indicator images. These files should be stored as a .gif or .jpg and you must know where they are located on the controller.

  The most common place to store these images is on FR:. If these are stored on MC:, then the memory card must be inserted to view the Visual Diagnostic screen properly.

  The screen area allocated for the background image is 630 pixels wide by 330 pixels high. For maximum performance, the background image should not be larger than this.

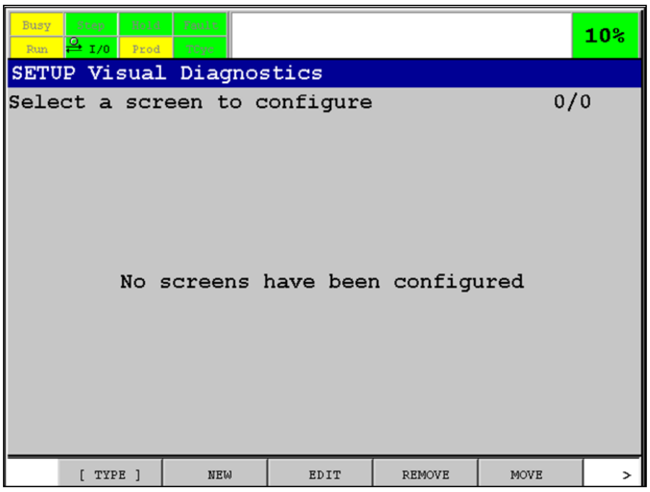
Steps

1 Press [MENU] key.

2 Select SETUP.

3 Press F1, [TYPE].

4 Select Visual Diagnostics. If this is the first time you are creating a Visual Diagnostic screen, the following screen will be displayed.

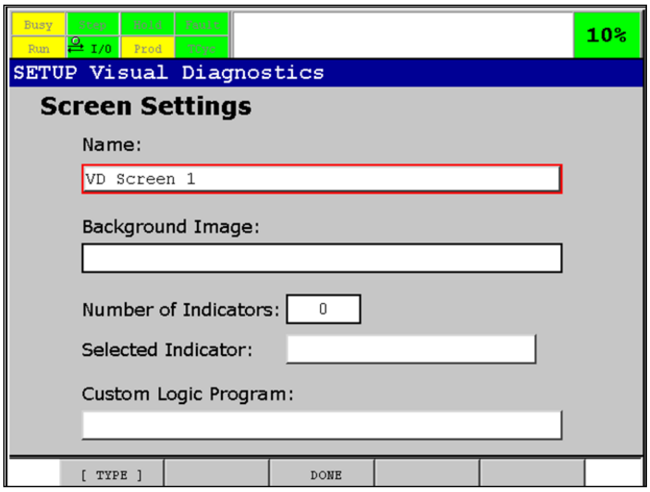


```
SETUP Visual Diagnostics
Select a screen to configure                0/0

        No screens have been configured
```

Fig. 51.1 (a) Select a Screen to Configure

If this is not the first time you have created a Visual Diagnostic screen, thumbnails of Visual Diagnostic screens currently configured on the controller are listed. To edit an existing screen, select the screen and press F3, EDIT, go to Step 6.

5 To create new screen, press F2, NEW. You will see a screen similar to the one shown in Fig.51.2(b). Refer to Table 51.1(a) and Table 51.1(b) to set up the new screen.



SETUP Visual Diagnostics

Screen Settings

Name:

VD Screen 1

Background Image:

Number of Indicators: 0

Selected Indicator:

Custom Logic Program:

Fig. 51.1 (b) Screen Settings Page

The red box on the screen indicates the selected field. You can select a field using the arrow keys. If you have a touch screen-enabled *i*Pendant, you can select a field by touching it. The function keys are adjusted to reflect operations that can be performed on the selected field. F3 is always DONE which, along with PREV, returns you to the previous SETUP screen.

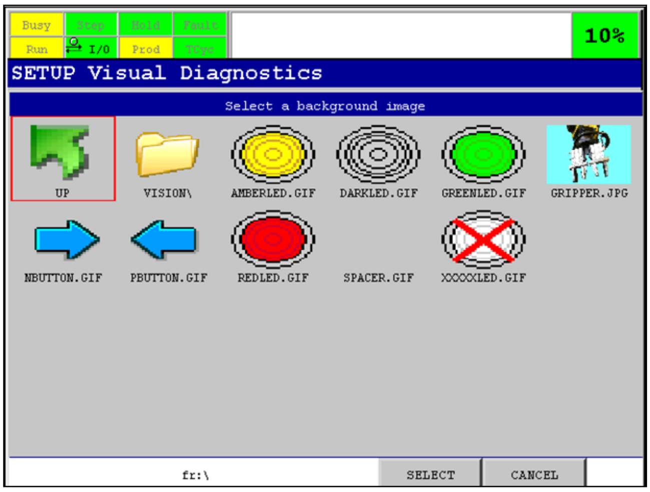
Table 51.1 (a) SETUP Visual Diagnostics Screen Settings Setup Items

Item	Description
Name	The item is the text that shows up on the SETUP and STATUS list screens so it must be unique. It can be from 1 to 40 characters and it can contain spaces. You can modify the name directly on the SETUP Visual Diagnostics Screen Settings screen.
Background Image	The large picture that covers most of the Visual Diagnostic screen is called the background image. It is stored as a file. When set, the file name is shown on the SETUP Visual Diagnostics Screen Setting screen.
Number of Indicators	This item specifies the number of indicators that have been set up. You cannot modify the number directly. Only the function keys are active when you are on this field.
Selected Indicator	Only one indicator can be configured at a time. The Selected Indicator field specifies which one is currently selected.

Item	Description
Custom Logic Program	You can identify a program that will run each time the Visual Diagnostic screen is refreshed. Use this feature if complex logic is required to determine the state of an indicator. A typical custom logic program will combine the status of various elements in the controller and set the indicators state control elements accordingly. A custom logic program can be implemented in either TPP or KAREL. It must not contain any motion statements. Use the following guidelines to create a custom logic program: • The program must exist before setting this field but you can modify it later to change the logic. The most recent version of the program is run each time the screen is update. • If it is a teach pendant program, the attribute should be set as follows: BUSY_LAMP_OFF = 3; ABORT_REQUEST = 3; PAUSE_REQUEST = 7; DEFAULT_GROUP = *, *, *, *, *, *, *, * The BUSY_LAMP_OFF setting is recommended so that the lamp which indicates the program running accurately reflects the status of the foreground program only. The ABORT_REQUEST setting is recommended so that the custom logic program continues to run eve if the user aborts the foreground program. The PAUSE_REQUEST setting is recommended so that the custom logic program continues to run eve if the user pauses the foreground program. The DEFAULT_GROUP setting is recommended so that the custom logic program runs in background. These settings can only be made by editing the TP program in the ASCII format (.LS). In order to load the ASCII program, ASCII Program Loader option (A05B-2600-R796) is required. • The program must run quickly and terminate. If the program takes more than 1 second to complete an warning alarm will be posted but Visual Diagnostics will proceed. One second is very long. A typical program should complete in less than a millisecond.

6 To change the VD screen name,
   a. Select the Name field, and press [ENTER] key.
   b. Type the name of the screen, and press OK.

7 To select the background image,
   a. Select the Background Image field and press F4, [CHOICE].
   See the screen shown in Fig. 51.1 (c) for an example.

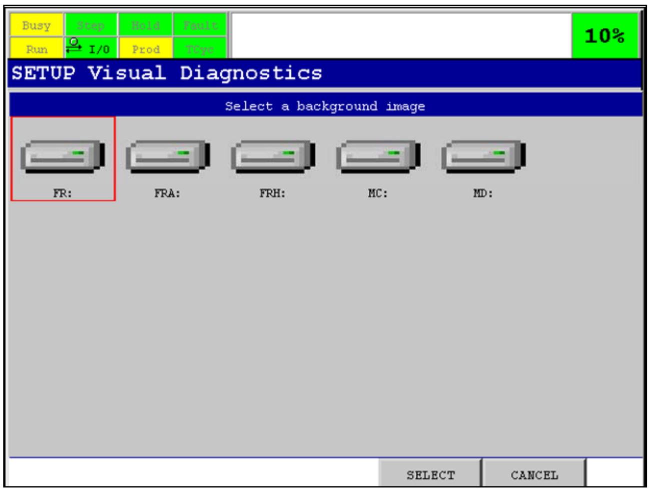


SETUP Visual Diagnostics

Select a background image

Fig. 51.1 (c) Selecting an Image File

b. The red box on the screen indicates the item with focus. Use the arrow keys to move the red box when more than one item is displayed. Press ENTER or F4 SELECT to select the item with focus. If you have a touch screen *i*Pendant, touching the item gives it focus and selects it in one step. Selecting the green arrow displays the images and folders in the parent folder. The top most parent is a list of devices. See Fig. 51.1 (d). Selecting a folder or a device displays the folders and image files within it. Selecting an image file returns that name to the field being edited in the Visual Diagnostics setup screen.



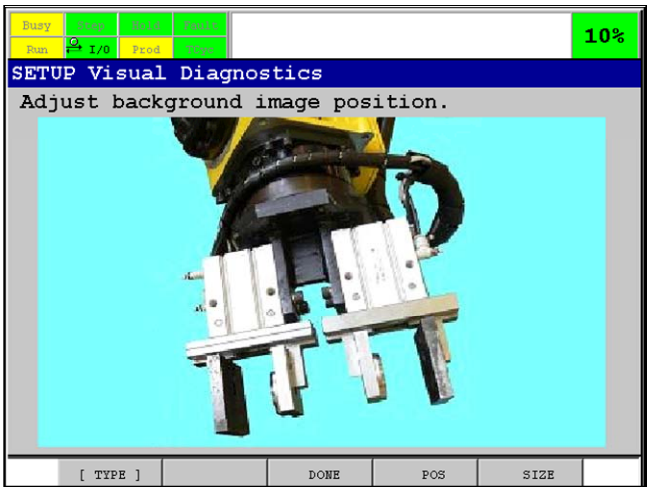
SETUP Visual Diagnostics

Select a background image

Fig. 51.1 (d) File Devices

8 To adjust background image,

a. Select the Background Image field and press F2, ADJUST. The selected graphic will be displayed in the center of the screen as shown in Fig. 51.1 (e).



SETUP Visual Diagnostics

Adjust background image position.

Fig. 51.1 (e) Adjust

b. F4, POS and F5, SIZE act like sticky radio buttons. Only one or the other is active at a time. When F4, POS is active, the arrow keys reposition the indicator image. When F5, SIZE is active, the up and down arrow keys grow and shrink the height of the image. The right and left arrow keys grow and shrink the width.

c. To position or resize the graphic in larger increments, press [SHIFT] key and then use the arrow keys.

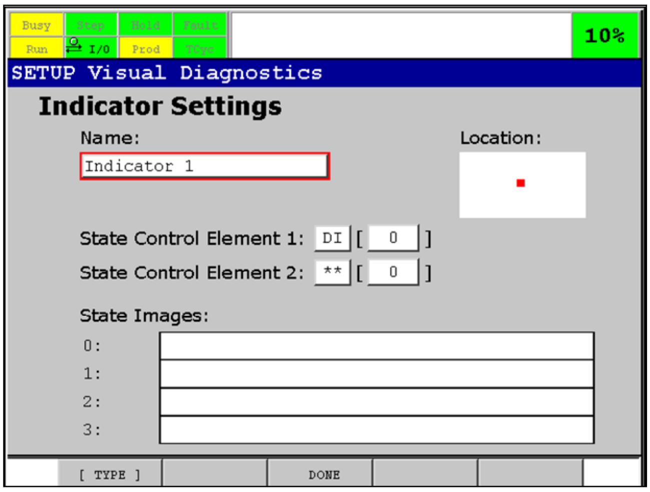
d. To complete the operation, press F3, DONE or [PREV] key.

9 To add in indicator, select the Number of Indicators field and press F2, NEW. A new indicator with default settings will be created and you will immediately be placed into the edit mode as described below and shown in Fig. 51.1 (f).

10 To edit an existing indicator,

a. Select the Selected Indicator field and press F2, EDIT.

b. Only one indicator can be edited at a time. If the name of the indicator you want to edit is not the one displayed, press F4, [CHOICE] and select it from the displayed list.
Indicator editing is done on a screen similar to the one shown in Fig. 51.1 (f).
Refer to Table 51.2(b) for information on setting up items on the Indicator Setting screen.



SETUP Visual Diagnostics

Indicator Settings

Name: Indicator 1

Location:

State Control Element 1: DI [0]

State Control Element 2: ** [0]

State Images:

0:

1:

2:

3:

Fig. 51.1 (f) Indicator Settings Page

Table 51.1 (b) Setup Visual Diagnostics Indicator Setup Items

Item	Description
Name	This item specifies the name of the indicator. It provides you with a way to reference the indicator for setup. It is not displayed on the Visual Diagnostic STATUS screen.
Location	The Location field shows a graphic that provides a rough idea where the indicator is located.
State Control Elem 1	This item specifies the first element that controls the indicator. It can be an I/O signal, numeric register, or vision register.
State Control Elem 2	This item specifies an optional second element to combine with the first one to control the indicator.
State Images	These are the images to display when the State Control Element(s) evaluate to the corresponding number.

11 To name the indicator,

a. Select the Name field, and press ENTER.

b. Type in the new indicator name and press OK.

12 To adjust the location of the indicator on the graphic,

a. Select the Location field and press F2, ADJUST.
The background image will be displayed with a small red box placed on it showing where the indicator is located. See the figure below.

b. Use the arrow keys to move the small red square to the location you want. If you have a touch screen, you can touch on a new location. Press F3, DONE or PREV to complete the operation.

13 To change the State Control Element type,

a. Select the left field in State Control Element line, and press F4 [CHOICE].
Each State Control Element line has two fields. The type field is displayed as one or two characters to the left of the left bracket. The index field is between the brackets.
Valid types for this field are:

- Boolean I/O - DI, DO, F, RI, RO, SI, SO, UI, UO, WI, WO
- Numeric I/O - AI, AO, GI, GO
- Data Register - R
- *i*RVision Register – VR
- Not used - ** (Only valid for SCE2)

b. Select a type, and press [ENTER] key.

14 To change the State Control Element index,

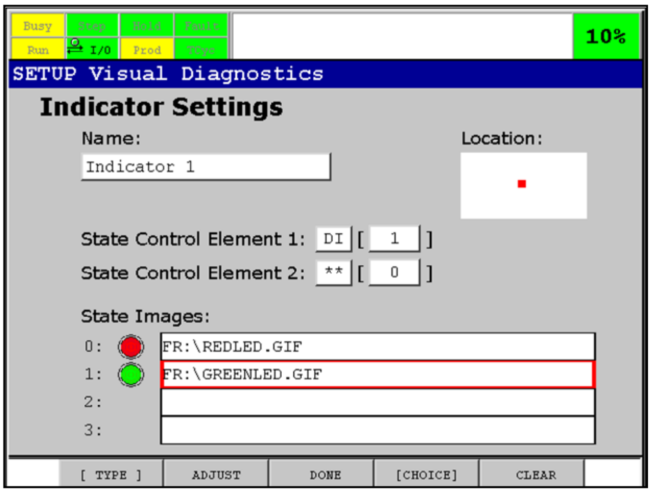
a. Select the right field and press [ENTER] key.

b. Enter the index number directly. No validation is done at this time. An invalid type or index is detected only when the Visual Diagnostic screen is viewed in the STATUS menu (refer to Section 51.4).

15 To associate an image to a state,

a. Select the image name field to the right of the desired state number and press F4, [CHOICE].

b. Use the file thumbnail browser to select the image. See Fig. 51.1 (g) for an example. Fig. 51.1 (g) shows the associated images to states 0 and 1.



SETUP Visual Diagnostics

Indicator Settings

Name: Indicator 1

Location:

State Control Element 1: DI [1]

State Control Element 2: ** [0]

State Images:

0: FR:\REDLED.GIF

1: FR:\GREENLED.GIF

2:

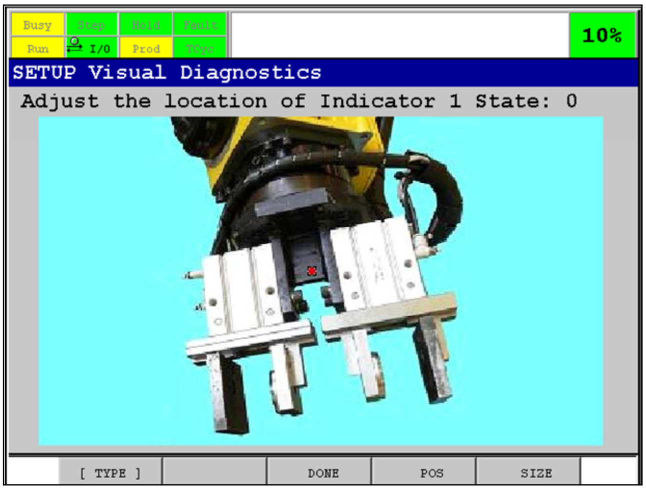
3:

Fig. 51.1 (g) Indicator Settings

16 To make fine position and scale adjustments to the state image,

a. Select the State Image field and Press F2, ADJUST.

b. The background image with only the currently selected indicator state image overlaid on top will be displayed. See Fig. 51.1 (h).



SETUP Visual Diagnostics

Adjust the location of Indicator 1 State: 0

Fig. 51.1 (h) Adjust the Location of LED Indicator State: 0

The position adjustment is relative to the grid location chosen for the indicator. These adjustments apply to the specific indicator state on the specific Visual Diagnostic screen being set up. They do not alter the indicator image file itself.

c. F4, POS and F5, SIZE act like sticky radio buttons. Only one or the other is active at a time. When F4, POS is active, the arrow keys reposition the indicator image. When F5, SIZE is active, the up and down arrow keys grow and shrink the height of the indicator image. The right and left arrow keys grow and shrink the width. To position or resize the graphic in large increments, press SHIFT and then use the arrow keys.

d. Press F3, DONE or PREV to complete the operation.

17 When you are finished setting up the indicators on the screen, press F3, DONE or PREV. This will return you to the Screen Setting page on which the following procedures are performed.

18 To remove the selected indicator from a screen,

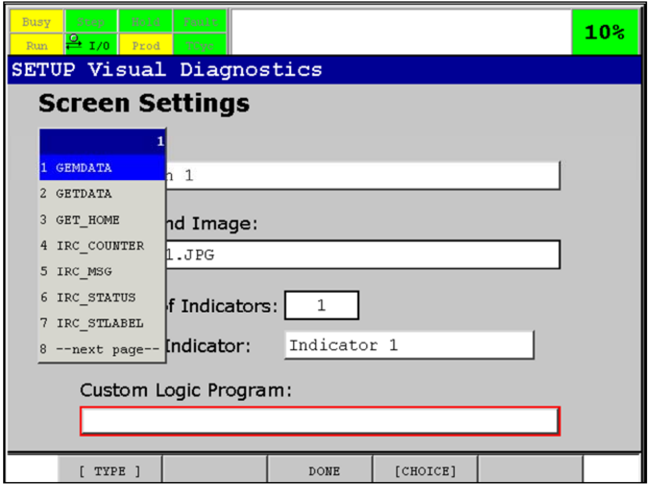
a. Select the Selected Indicator field and press F5, REMOVE.

b. A confirmation box will be displayed so that you are aware that the indicator will be permanently removed by this function.

19 To assign a custom logic program,

a. Select the Custom Logic Program field and press F4, [CHOICE]. This will display the list of programs currently loaded on the controller.

b. Select the program you want from this list, and press ENTER. See the Fig.51.1 (i) for an example.



SETUP Visual Diagnostics

Screen Settings

1
1 GEMDATA
2 GETDATA
3 GET_HOME
4 IRC_COUNTER
5 IRC_MSG
6 IRC_STATUS
7 IRC_STLABEL
8 --next page--

n 1

nd Image:

1.JPG

f Indicators: 1

Indicator: Indicator 1

Custom Logic Program:

Fig. 51.1 (i) Custom Logic Program Field

20 To clear the Custom Logic Program assignment, select the Custom Logic Program field and press F5, CLEAR.

51.2 MANAGING SCREENS

You can perform functions such as remove, move, copy, and load a new screen.

If the selected item is currently being displayed in any other *i*Pendant pane, the EDIT and REMOVE function keys will not let you operate on it.

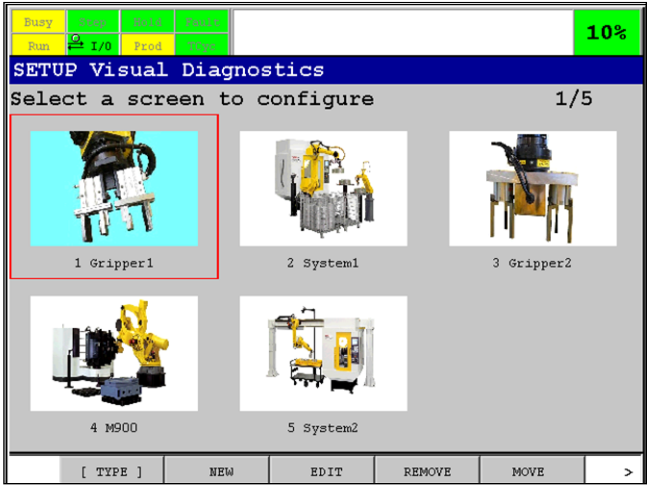
Procedure 51-2 Managing Screens

Condition

- Visual Diagnostic screens have already been set up.

Step

1 Press [MENU] key.
2 Select SETUP.
3 Press F1, [TYPE].
4 Select Visual Diagnostics. The following screen will be displayed.



SETUP Visual Diagnostics

Select a screen to configure 1/5

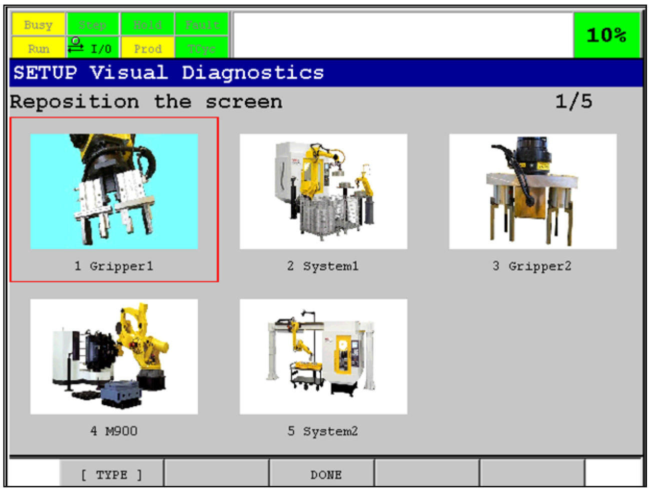
1 Gripper1
2 System1
3 Gripper2
4 M900
5 System2

Fig.51.2(a) SETUP Visual Diagnostics

5 To remove a screen, press F4, REMOVE. A confirmation box will be displayed so that you are aware that the screen will be permanently removed by this function.

6 To reposition a screen,

a. Press F5, MOVE. The following screen will be displayed.



SETUP Visual Diagnostics

Reposition the screen 1/5

Fig.51.2(b) Reposition the Screen

b. Use the arrow keys to move the selected screen to its new position.

c. Press F3, DONE to finish the operation and return to the screen selection page.

7 To make a new screen as an exact copy of the currently selected screen but with a different screen name,

a. Press >, NEXT, and then press F2, COPY. A new screen will be created and the Screen Settings page shown so that you can edit the new screen.

b. Use Procedure 51-1 to edit the screen.

8 To load Visual Diagnostic screens developed on another controller onto this controller,

a. Press >, NEXT and then press F3, LOAD.

b. Using the file browser, find and select the file named “vmscreens.xml” on any file device. When selected, the system merges the screens defined in it with the screens that currently reside on the controller. The file named “vmscreens.xml” is saved by the file backup operation in FILE screen. When the file backup operation is done in FILE screen, if “Application” or “All of above” is selected in the backup menu, the file named “vmscreens.xml” is saved to the designated device.

c. If a duplicate screen name is encountered, you are prompted with the option of replacing or keeping the one resident on the controller.

51.3 VIEWING VISUAL DIAGNOSTIC SCREENS

Use Procedure 51-3 to view Visual Diagnostic Screens.

Procedure 51-3 Viewing Visual Diagnostic Screens

Condition

- The Visual Diagnostic screen you want has already been set up.

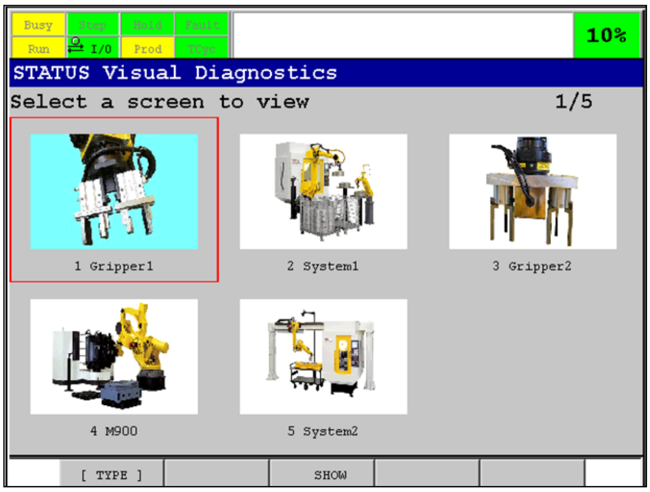
Step

1 Press [MENU] key.

2 Select SETUP.

3 Press F1, [TYPE].

4 Select Visual Diagnostics. The following screen will be displayed.



STATUS Visual Diagnostics

Select a screen to view 1/5

Fig. 51.3 (a) Select a Screen to View

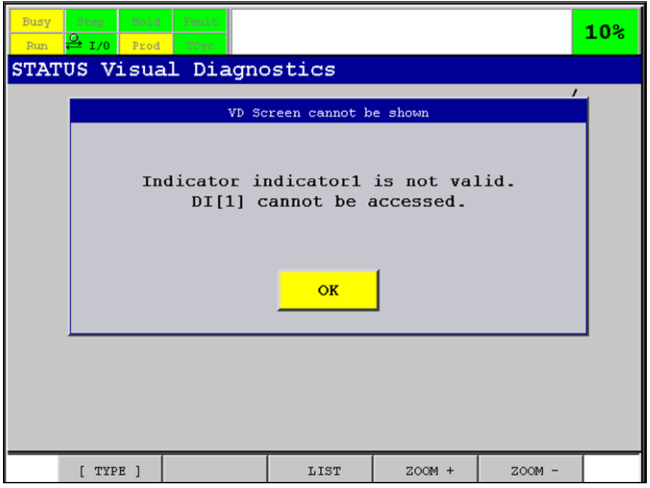
Use this menu item to display a Visual Diagnostic screen. Thumbnails of the set of screens configured on the controller are shown on the list selection screen.

The red box on the screen indicates the selected screen. You can select a screen using the arrow keys. If you have a touch screen-enabled *i*Pendant, you can select a screen by touching it. The numbers in the upper right hand corner indicate how many screens are available and which is currently selected.

5 To display a screen, select it and press F3, SHOW.

If the selected item is currently being edited in any other *i*Pendant pane, F3, SHOW will not let you view it.

6 If the screen you have selected has an invalid signal set, the following screen will be displayed.



STATUS Visual Diagnostics

VD Screen cannot be shown

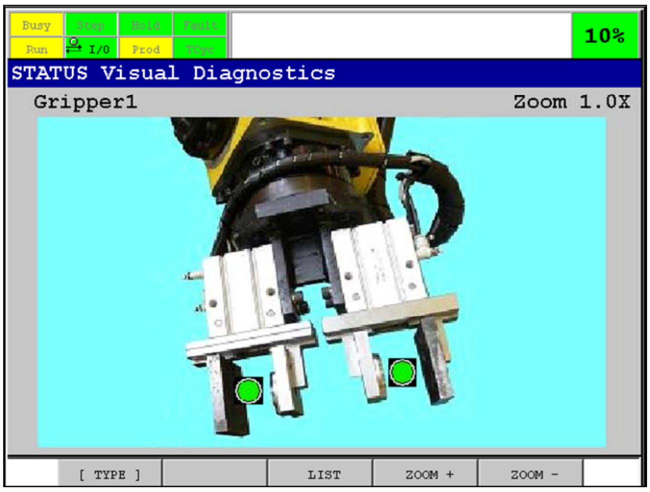
Indicator indicator1 is not valid.
DI[1] cannot be accessed.

Fig.51.3(b) Invalid Indicator

a. Press OK.

b. Go to the SETUP menu to redefine the signal for that indicator. See Procedure 51-1.

7 After you select a Visual Diagnostic screen, the following screen will be displayed.



STATUS Visual Diagnostics

Gripper1 Zoom 1.0X

Fig.51.3(c) Visual Diagnostics Screen Display

The top line displays the title of the screen. In this example, the title is Gripper1.
Adjustments (since controller power up) to the magnification and picture center are remembered for each Visual Diagnostic screen. So, if you leave the screen and then return, the view is restored.

8 To increase magnification of the image,
  a. Press F4, ZOOM+.
  b. The magnification is increased keeping the point of the image in the center of the display fixed in place.

9 To decrease magnification of the image,
  a. Press F5, ZOOM-. The point of the picture in the center of the display is kept in place until one of the edges of the image touches the edge of the display area. The image will then be shifted in that direction to keep the image edge touching the display edge.
  b. Magnification will not be reduced below 100%.
  c. F2, FULL, restore the image to full view.

10 To adjust how the image is centered in the display when the magnification is greater than 100%,
  a. Press the right and left arrow keys to shift the view to the right and left.
  b. Press the up and down arrow keys to shift the view up and down.
  c. To move the view in larger increments, hold [SHIFT] key while pressing the arrow key.

11 To display the Visual Diagnostic Screen list selection page, press F3, LIST. This key has the same function as [PREV] key.

52 PDF VIEWER FUNCTION

The iPendant PDF Viewer function allows viewing of PDF documents in proprietary file formats.
The following file formats are supported:

Adobe® Acrobat® Portable Document Format (PDF) v1.3 and later.

To use this function, " 4D Graphics (R764)" option is required.

52.1 OPENING A PDF DOCUMENT

PDF documents can be opened on any local R-30*i*B storage device.

Procedure 52-1 Open PDF Document

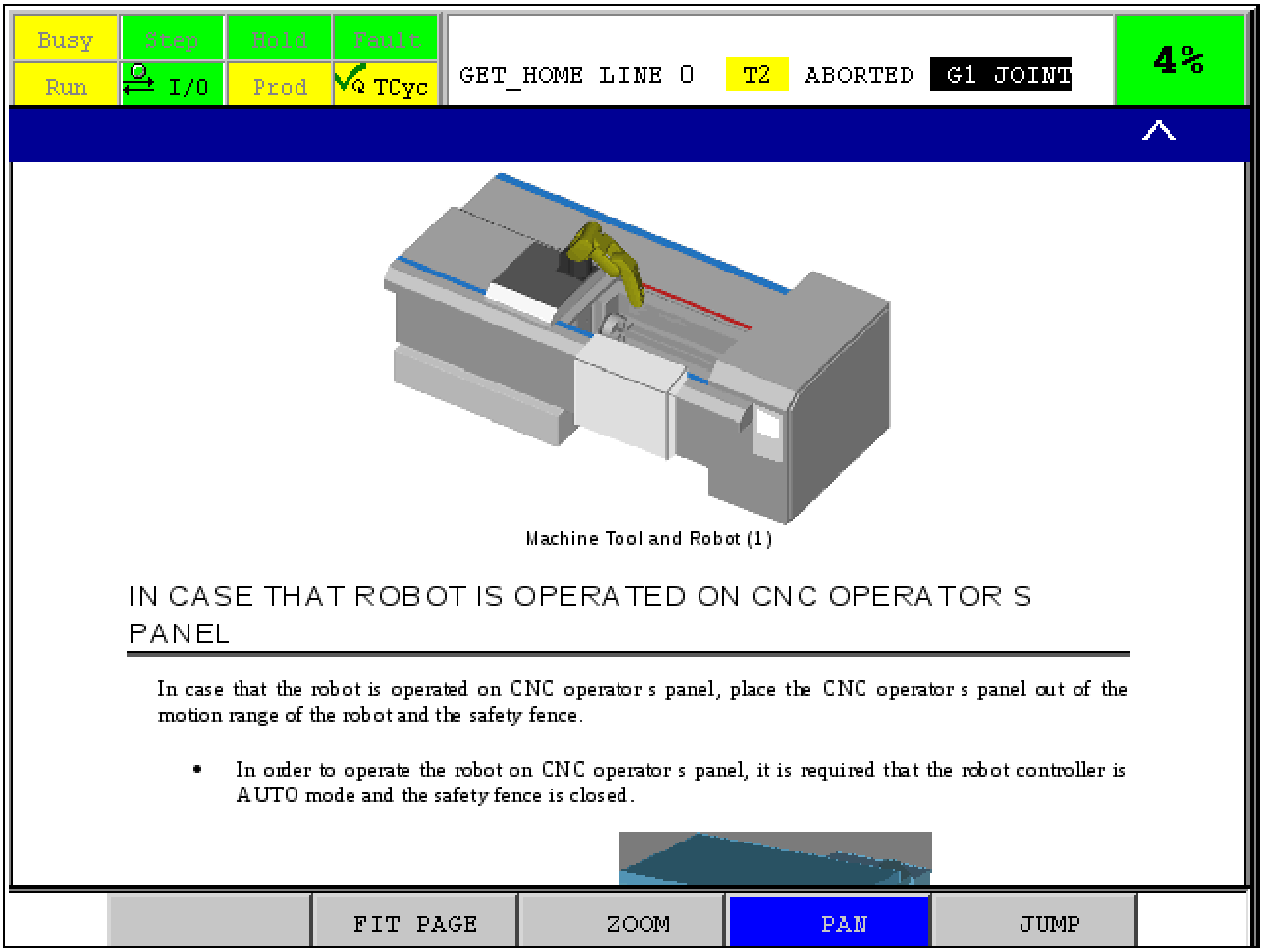
1 Press [MENU] key.
2 Select FILE.
3 Press F5, [UTIL]
4 Select Set Device.
5 Select a device.
6 Press F2, [DIR]
7 Select a filter.
8 Select the file to view.
9 Press [ENTER] key.

⚠ **WARNING**
It is required to change the display type of the file screen to “Wide” before selecting file name when pdf file size is larger than 10 Mbyte. Press NEXT key, press F5[VIEW] and select “Wide” to change to “Wide” screen.

⚠ **WARNING**
It may take a lot of time to display or scroll when pdf file size is very large.

52.2 NAVIGATING WITH THE PDF VIEWER

Fig. 52.2 shows a typical single pane view of a document open in the PDF Viewer.



IN CASE THAT ROBOT IS OPERATED ON CNC OPERATOR S PANEL

In case that the robot is operated on CNC operator s panel, place the CNC operator s panel out of the motion range of the robot and the safety fence.

- In order to operate the robot on CNC operator s panel, it is required that the robot controller is AUTO mode and the safety fence is closed.

Fig. 52.2 PDF Viewer

Three function keys are available when the PDF Viewer has focus.

Table 52.2 PDF Viewer Function Keys

Function Key	Description
FIT PAGE	Zoom so that the entire page is visible within the PDF Viewer window. this mode allows quick navigation between pages of a document.
ZOOM	The up and down arrow key will increase and decrease the zoom level respectively.
PAN	The arrow keys will pan the display. Touch screen gestures will always pan the display.
JUMP	Table of contents will be displayed and can jump to each page if PDF file has the table of contents.

53 HELP AND DIAGNOSTICS DISPLAY

You can use the DIAG/HELP key to display help for the data displayed in the current window on *i*Pendant or diagnostics for an alarm.

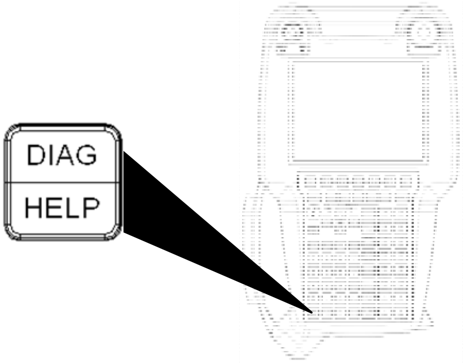


Fig. 53 DIAG/HELP key

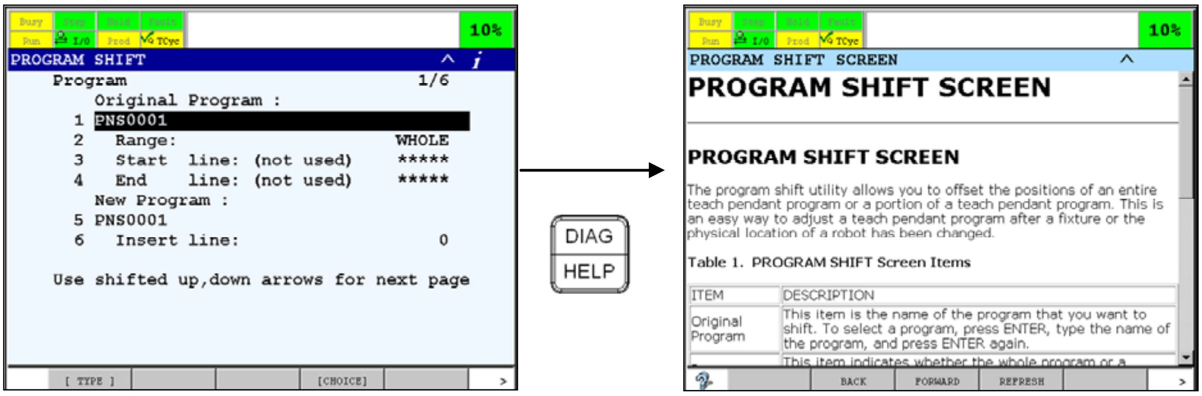
- If DIAG/HELP key is pressed, help for the current screen will be displayed.
- If DIAG/HELP key is pressed while holding down [SHIFT] key, diagnostics for an active alarm or the specified alarm is displayed.

To use these functions, the following software options are required.

● Help Display:	Online Help	(A05B-2600-R664)
● Diagnostics Display:	Alarm Cause/Remedy	(A05B-2600-R665)

53.1 ONLINE HELP FUNCTION

If [DIAG/HELP] key on the teach pendant is pressed, the help to explain the operation in the current screen is displayed. For example, if [DIAG/HELP] key is pressed while the program shift screen is displayed, the help for the program shift screen is displayed as follows.



```
PROGRAM SHIFT
    Program                            1/6
      Original Program :
     1 PNS0001
     2   Range:                      WHOLE
     3   Start  line:  (not used)    *****
     4   End    line:  (not used)    *****
      New Program :
     5 PNS0001
     6   Insert line:                    0

    Use shifted up,down arrows for next page
```

PROGRAM SHIFT SCREEN

PROGRAM SHIFT SCREEN

PROGRAM SHIFT SCREEN

The program shift utility allows you to offset the positions of an entire teach pendant program or a portion of a teach pendant program. This is an easy way to adjust a teach pendant program after a fixture or the physical location of a robot has been changed.

Table 1. PROGRAM SHIFT Screen Items

ITEM	DESCRIPTION
Original Program	This item is the name of the program that you want to shift. To select a program, press ENTER, type the name of the program, and press ENTER again.
	This item indicates whether the whole program or a

Fig. 53.1 Help for Program Shift Screen

To return to the previous screen, press [PREV] key or DIAG/HELP key. You can also display the previous screen by pressing DIAG/HELP key with [SHIFT] key.

53.2 ALARM CAUSE/REMEDY DISPLAY FUNCTION

If [DIAG/HELP] key on the teach pendant is pressed with [SHIFT] key, the alarm cause and remedy can be displayed as the diagnostic information. According to the displayed screen and alarm occurring status, the displayed contents of the alarm cause and remedy differ as follows.

1 In case that the alarm screen (the active alarm screen or the alarm log screen) is displayed, the cause and remedy for the specified alarm by the cursor in the alarm screen is displayed.
2 In case that the active alarm screen is displayed, if no alarm is displayed in the screen, the cause and remedy for the selected alarm in the alarm list is displayed.
3 In case that the screen other than the alarm screen is displayed, when an alarm occurs, the cause and remedy for the active alarm is displayed.
4 In case that the screen other than the alarm screen is displayed, when no alarm occurs, the cause and remedy for the selected alarm in the alarm list is displayed.

To return to the previous screen, press [PREV] key or [DIAG/HELP] key. You can also display the previous screen by pressing [DIAG/HELP] key with [SHIFT] key.

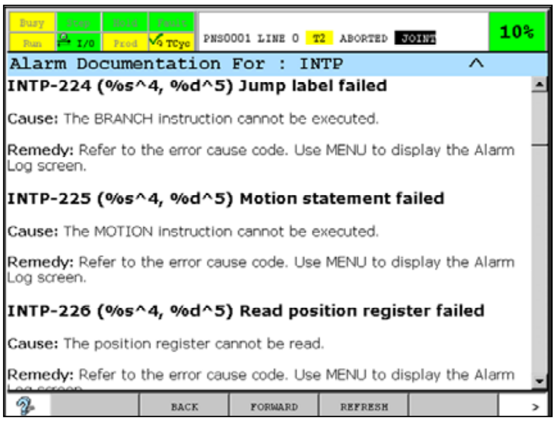
Procedure 53-1 Diagnostics display when alarm screen is displayed.

Conditions

- The alarm screen (the active alarm screen or the alarm log screen) is displayed.
- The alarm is displayed in the screen.

Steps

1 Move the cursor to the alarm that you want to display the cause and remedy.
In case that only one alarm is displayed in the active alarm screen, the cursor cannot be moved. The cause and remedy for the displayed alarm in the active alarm screen is displayed.
2 Press [DIAG/HELP] key while holding down [SHIFT] key.
3 The cause and remedy for the specified alarm is displayed.



Alarm Documentation For : INTP

INTP-224 (%s^4, %d^5) Jump label failed

Cause: The BRANCH instruction cannot be executed.

Remedy: Refer to the error cause code. Use MENU to display the Alarm Log screen.

INTP-225 (%s^4, %d^5) Motion statement failed

Cause: The MOTION instruction cannot be executed.

Remedy: Refer to the error cause code. Use MENU to display the Alarm Log screen.

INTP-226 (%s^4, %d^5) Read position register failed

Cause: The position register cannot be read.

Remedy: Refer to the error cause code. Use MENU to display the Alarm

Fig. 53.2 (a) Alarm Cause and Remedy Display

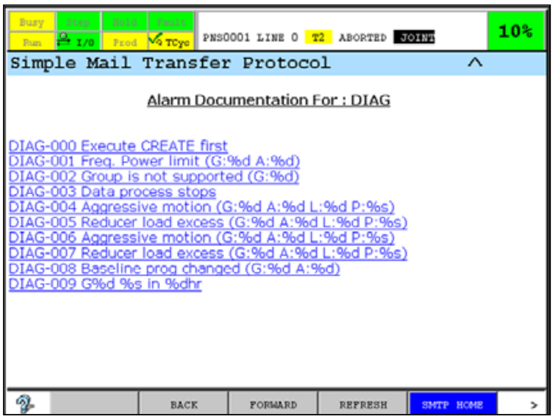
Procedure 53-2 Diagnostics display when the active alarm screen is displayed and no alarm is displayed in the screen.

Conditions

- The active alarm screen is displayed.
- No alarm is displayed in the screen.

Steps

1 Press [DIAG/HELP] key while holding down [SHIFT] key.
2 The list screen for DIAG alarm is displayed as follows.



PNS0001 LINE 0 T2 ABORTED JOINT

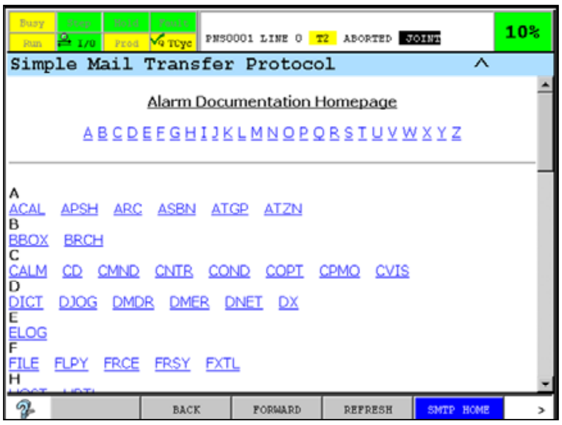
Simple Mail Transfer Protocol

Alarm Documentation For : DIAG

DIAG-000 Execute CREATE first
DIAG-001 Freq. Power limit (G:%d A:%d)
DIAG-002 Group is not supported (G:%d)
DIAG-003 Data process stops
DIAG-004 Aggressive motion (G:%d A:%d L:%d P:%s)
DIAG-005 Reducer load excess (G:%d A:%d L:%d P:%s)
DIAG-006 Aggressive motion (G:%d A:%d L:%d P:%s)
DIAG-007 Reducer load excess (G:%d A:%d L:%d P:%s)
DIAG-008 Baseline prog changed (G:%d A:%d)
DIAG-009 G%d %s in %dhr

Fig. 53.2 (b) List screen for DIAG alarm

3 Press SMTP HOME (F5). The following screen will be displayed.



PNS0001 LINE 0 T2 ABORTED JOINT

Simple Mail Transfer Protocol

Alarm Documentation Homepage

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A
ACAL APSH ARC ASBN ATGP ATZN
B
BBOX BRCH
C
CALM CD CMND CNTR COND COPT CPMO CVIS
D
DICT DJOG DMDR DMER DNET DX
E
ELOG
F
FILE FLPY FRCE FRSY FXTL
H

Fig. 53.2 (c) Alarm List screen

4 You can display the list of the alarm that you want to display the cause and remedy by selecting facility name in this screen.

5 Select the alarm to display the cause and remedy in the alarm list screen.

Procedure 53-3 Diagnostics display when the screen other than alarm screen is displayed and an alarm occurs.

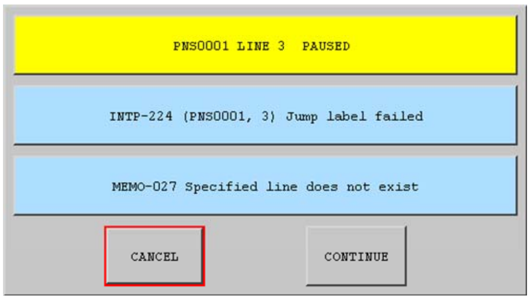
Conditions

- The screen other than the alarm screen is displayed.
- An alarm occurs.

Steps

1 Press [DIAG/HELP] key while holding down [SHIFT] key.

2 The following popup screen will be displayed.



PNS0001 LINE 3 PAUSED

INTP-224 (PNS0001, 3) Jump label failed

MEMO-027 Specified line does not exist

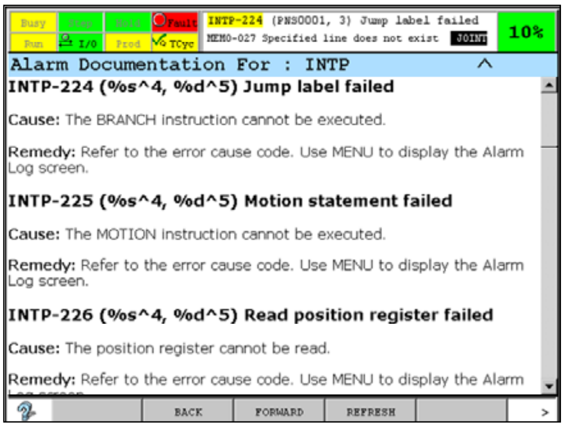
Program Status

The most recent alarm

Cause Code

Fig.53.2(d) Popup screen (When an alarm occurs)

3 In case that the “CONTINUE” button is pressed, the cause and remedy for the active alarm is displayed as follows. In case that the “CANCEL” button is pressed, the popup screen disappears and the previous screen will be displayed.



INTP-224 (PNS0001, 3) Jump label failed
MEMO-027 Specified line does not exist

Alarm Documentation For : INTP

INTP-224 (%s^4, %d^5) Jump label failed

Cause: The BRANCH instruction cannot be executed.

Remedy: Refer to the error cause code. Use MENU to display the Alarm Log screen.

INTP-225 (%s^4, %d^5) Motion statement failed

Cause: The MOTION instruction cannot be executed.

Remedy: Refer to the error cause code. Use MENU to display the Alarm Log screen.

INTP-226 (%s^4, %d^5) Read position register failed

Cause: The position register cannot be read.

Remedy: Refer to the error cause code. Use MENU to display the Alarm

Fig.53.2(e) Alarm Cause and Remedy Display

Procedure 53-4 Diagnostics display when the screen other than alarm screen is displayed and no alarm occurs.

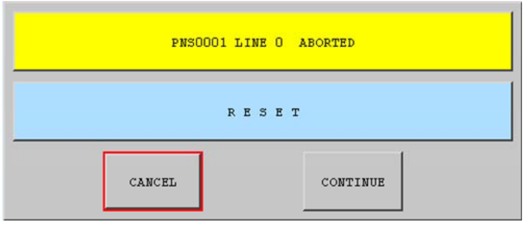
Conditions

- An alarm does not occur.
- The screen other than the alarm screen is displayed.

Steps

1 Press [DIAG/HELP] key while holding down [SHIFT] key.

2 The following popup screen will be displayed.



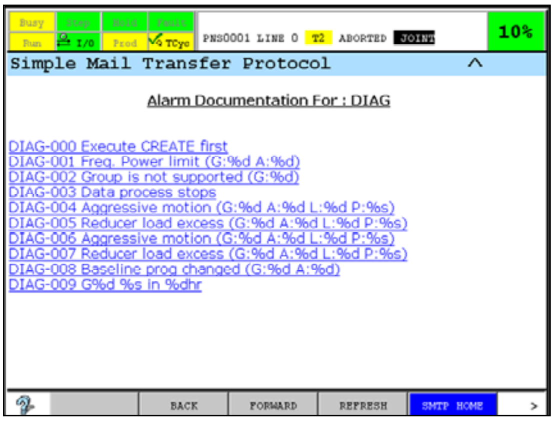
PNS0001 LINE 0 ABORTED

RESET

CANCEL CONTINUE

Fig.53.2(f) Popup screen (When no alarm occurs)

3 In case that the “CONTINUE” button is pressed, the list screen for DIAG alarm is displayed. Please refer to Procedure 53-2 to display the alarm cause and remedy from this screen. In case that the “CANCEL” button is pressed, the popup screen disappears and the previous screen will be displayed .



PNS0001 LINE 0 ABORTED

Simple Mail Transfer Protocol

Alarm Documentation For : DIAG

DIAG-000 Execute CREATE first
DIAG-001 Freq. Power limit (G:%d A:%d)
DIAG-002 Group is not supported (G:%d)
DIAG-003 Data process stops
DIAG-004 Aggressive motion (G:%d A:%d L:%d P:%s)
DIAG-005 Reducer load excess (G:%d A:%d L:%d P:%s)
DIAG-006 Aggressive motion (G:%d A:%d L:%d P:%s)
DIAG-007 Reducer load excess (G:%d A:%d L:%d P:%s)
DIAG-008 Baseline prog changed (G:%d A:%d)
DIAG-009 G%d %s in %dhr

Fig.53.2(g) List screen for DIAG alarm

53.3 HELP/DIAGNOSTICS SCREEN

The operations in the online help screen and the alarm cause and remedy screen are as follows.

Change of the display by the link

By the selecting the link, you can display the figure, the table, or the other page specified by the link.

- To highlight the link that you want to select, use the up and down cursor keys.
- If the teach pendant equips the touch panel, touch the link to select.

Scroll of the screen

If the screen exceeds the window, the scroll bar is displayed. Please scroll the screen by the following ways.

- To scroll the screen, press the up or down cursor key while holding down [SHIFT] key.
- If the teach pendant equips the touch panel, touch the scroll bar.

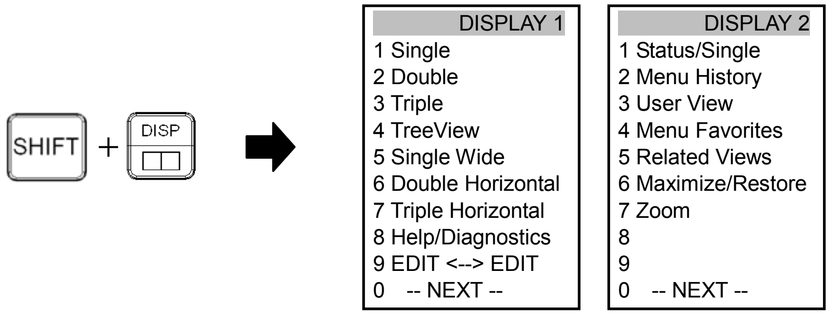
If the window on the teach pendant is divided, the Help/Diagnostics screen will be displayed as follows.

Table 53.3 Help/Diagnostics Display

Window mode	Display Location
Single	The Help/Diagnostics information replaces the current information in the window.
Status/Single	The Help/Diagnostics information replaces the current information in window on the right.
Double	The Help/Diagnostics information replaces the current information in the inactive window (window without focus).
Triple	The Help/Diagnostics information replaces the current information in the window without focus. For example in triple mode, the left window is window 1, the upper right window is window 2, and the lower right window is window 3. Then if the active window is window 2, the Help/Diagnostics information replaces the contents of window 3. If the active window is window 3, the Help/Diagnostics information replaces the contents of window 1, and so forth.

53.4 HELP/DIAGNOSTICS MENU

You can display an online help screen and an alarm cause and remedy screen by selecting an item in the Help/Diagnostics menu. The Help/Diagnostics menu will be displayed when the Help/Diagnostics item is selected in the DISPLAY menu that is displayed by pressing [DISP] key with [SHIFT] key.



(The items in the menu and the order of the items may differ depending on the software series and version.)

Fig. 53.4 (a) DISAPLY menu

When the Help/Display item is selected in the DISPLAY menu, the Help/Diagnostics menu will be displayed.

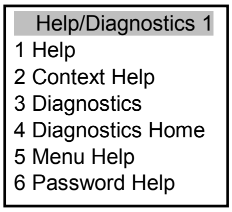


Fig. 53.4 (b) Help/Diagnostics menu

Table 53.4 describes the displayed screen by selecting each item is the Help/Diagnostics menu.

Table 53.4 Help/Diagnostics menu items

Item	Description
Help	This item displays help for the currently active window.
Context Help	This item displays context sensitive help, if exists, for the currently active window.
Diagnostics	When the alarm screen is displayed, this item displays the cause and remedy for the designated alarm in the alarm screen. When the active alarm screen is displayed, if there is no active alarm, the DIAG alarm list screen to display Diagnostics Home Page is displayed. When the screen other than the alarm screen is displayed, the alarm cause and remedy screen is not displayed by this item.
Diagnostics Home	This item displays the DIAG alarm list screen to display Diagnostics Home Page.
Menu Help	This item displays the menu tree and provides links to all the help files other than the current menu.

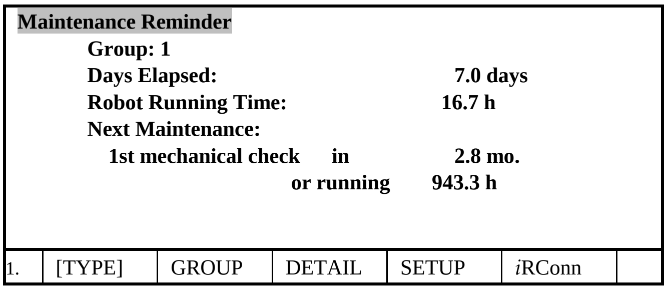
54 MAINTENANCE REMINDER

Maintenance Reminder reminds users of periodic maintenances or part replacement for a robot system. By notifying preconfigured and customized maintenances with TP warnings, output to DO, and Notification Icon (software versions 7DC3/10 and later), it helps to maintain with appropriate intervals. Maintenance items can be configured by the user. Some robot models have predefined items setup for recommended maintenance tasks (Required Maintenances). To use this function, Maintenance Reminder option (J771) is required. Also, when used in conjunction with *i*RConnect, the maintenances can be checked on mobile devices.

54.1 MAIN MENU

Main menu of Maintenance Reminder is displayed with following procedures.

1. Press [MENU] key.
2. Select “STATUS”.
3. Press F1, [TYPE].
4. Select “Reminder”.



```
Maintenance Reminder
          Group: 1
          Days Elapsed:                      7.0 days
          Robot Running Time:               16.7 h
          Next Maintenance:
             1st mechanical check    in      2.8 mo.
                           or running      943.3 h

1.  [TYPE]   GROUP   DETAIL   SETUP   iRConn
```

Table 54.1 items in main menu

Item	Description
Group	The group number, of which the maintenance items are currently shown.
Days Elapsed	It shows the days elapsed since the robot has started being used. It includes the time while the power is off. Adjusting the clock of the controller may result in increased elapsed time than in reality. All the groups in the controller share the same elapse time.
Robot Running Time	It shows the total time the robot has been moving. It does not include waiting time. Each group has individual counter.
Next Maintenance	Among enabled maintenance items, the item with shortest remaining time to the next maintenance will be shown with the remaining time. The items displayed will be limited to those setup with “Days Elapsed” or “Robot Running Time”.

If multiple groups exist, press F2 [GROUP] and the group number to change the group to show.

On software versions 7DC3/10 and later, pressing [NEXT] key then F3 “OUT LIST” key displays Monitored Out screen, F4 “RECORD” key displays Maintenance Record screen. They will be described in later sections.

54.2 SETUP

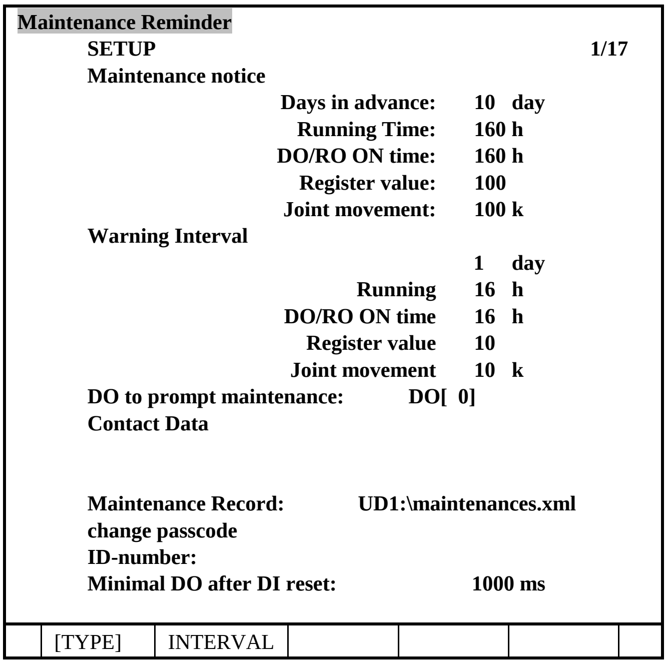
- There are settings common to groups and items, such as when to send reminder notice.
- There are settings individual to each maintenance items, such as maintenance intervals.
- If passcode, described in the next section, is set, showing setup menu asks for the passcode. Set the passcode for the cases such as if the maintenance personnel and operators differ.

54.2.1 Common Setting

This menu is for settings common among all the groups of the controller, such as the timing to output the maintenance reminder warning.

Setup menu is displayed with following procedures.

1. Display the main menu in the Section 54.1.
2. Press F4, [SETUP］.
3. If the passcode is set, the prompt line will ask “Please enter the passcode: “. Enter the passcode to display the setup menu.



```
Maintenance Reminder
          SETUP                                             1/17
          Maintenance notice
                             Days in advance:     10   day
                               Running Time:      160 h
                             DO/RO ON time:       160 h
                              Register value:     100
                             Joint movement:      100 k
          Warning Interval
                                                  1    day
                                     Running      16   h
                               DO/RO ON time      16   h
                              Register value      10
                              Joint movement      10   k
          DO to prompt maintenance:         DO[  0]
          Contact Data


          Maintenance Record:          UD1:\maintenances.xml
          change passcode
          ID-number:
          Minimal DO after DI reset:                1000 ms

 [TYPE]    INTERVAL
```

Table 54.2.1 Common setting items

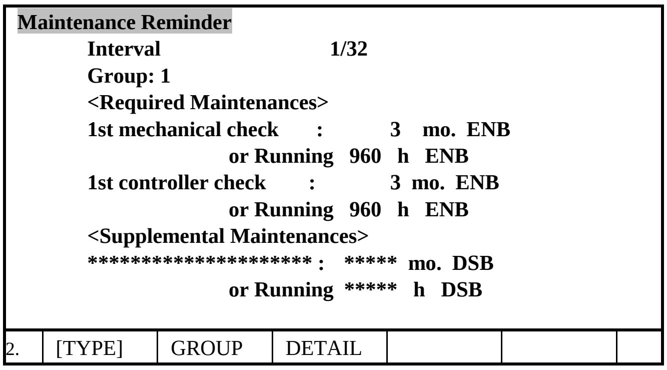
Item	Description
Maintenance notice	Set timing to post warnings and output DO. On Software versions 7DC3/10 and later, DO/RO ON time, Register value, and Joint movement can be set in addition to days and Running Time. When the remaining time or value to the next maintenance reaches the set time, warnings, DO, and Notification screen start to remind maintenances. The unit of Joint movement is thousand revolutions (kRev) for rotational axis, km for linear axis. If item specific setup, described in later section, is done, it will be used instead of this common setting.
Warning Interval	Set interval to post warnings after the maintenance notice time, explained above, has been reached. If item specific setup, described in later section, is done, it will be used instead of this common setting.
DO to prompt maintenance	Set DO to be output after the maintenance notice time, explained above, has been reached. DO will be ON when any item reaches maintenance notice time, and will stay ON until “DONE” operation, to be explained in the later section. Setting DO[0] will disable DO output. On software versions 7DC3/10 and later, item specific DO can be setup, which will be described in later section.
Contact Data	Any text up to 37 letters can be set for these 2 lines. When set, they will be displayed on the main menu in the Section 54.1. (They will not be displayed when no data is set.) Any data such as contact data for maintenance personnel can be set.
Maintenance Record	On software versions 7DC3/10 and later, Maintenance Record can be output to a file. This item sets destination file of the records. Maintenance Record will be output only after the file is created in Maintenance Record screen, which will be described in later section.

Item	Description
change passcode	Select this item and press ENTER to set passcode. (If passcode is already set,) After entering the current passcode, enter the new code twice to set the new passcode. Set "1111" to disable passcode protection. Do not forget the passcode, or settings cannot be changed and "DONE" operation, to be explained in the later section, cannot be executed.
ID-number	Any text up to 10 letters can be set as ID of the controller. When set, it will be displayed on the main menu in the Section 54.1. (It will not be displayed when no ID is set.) It can be used for any purpose, such as for maintenance management.
Minimal DO after DI reset	On software versions 7DC3/10 and later, maintenance items can be reset by DI, which will be described in later section. This item sets the minimal time length of DO output when DI reset is done. When checking the DO by external PLC and such, set the length to be longer than the cycle length of the checking. The actual time length of DO output may be longer depending on the processing status of the controller.

54.2.2 Maintenance Item Setting

Interval setting menu is displayed with following procedures.

1. Display setup menu in the Subsection 54.2.1.
2. Press F2 [INTERVAL］.



```
Maintenance Reminder
        Interval                    1/32
        Group: 1
        <Required Maintenances>
        1st mechanical check    :       3   mo.  ENB
                      or Running   960   h   ENB
        1st controller check    :       3  mo.  ENB
                      or Running   960   h   ENB
        <Supplemental Maintenances>
        ********************* :   *****  mo.  DSB
                      or Running  *****   h   DSB
```

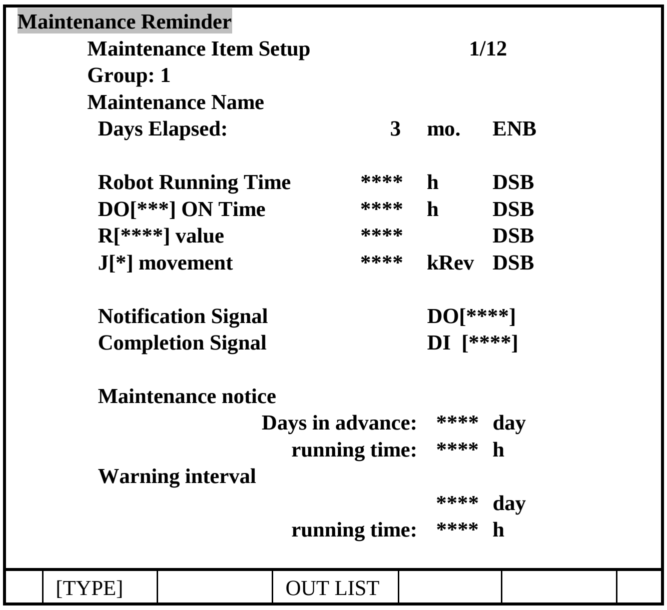
There are two types of maintenance item, Required Maintenances and Supplemental Maintenances. Required Maintenances are prepared for the robot models, and any maintenance can be added to Supplemental Maintenances. The configurability of two types differs, but they act in the same way after the setup.

Table 54.2.2(a) Maintenance Item settings

Item	Description
Group	The group number, of which the maintenance items are currently shown.
Name	The name of the maintenance item. The names of the Required Maintenances are fixed. The names of Supplemental Maintenances can be set freely, up to 21 letters.
Interval	The interval, with which the maintenance should be done. The intervals can be set for "Days Elapsed" and/or "Robot Running Time", "DO/RO ON Time", "Register value", or "Joint movement". The units of interval for "Days Elapsed" are days (day) or months (mo.) which can be switched by pressing F4/F5 keys while selecting the unit. The unit of interval for "Robot Running Time" and "DO/RO ON Time" is hours (h). The unit of interval for "Joint movement" is thousand revolutions (kRev) for rotational axis, or km for linear axis. Required Maintenances have the intervals setup. They can be shortened, but cannot be made longer.
ENB/DSB	Enable or disable the interval type of the maintenance item. Select the item and press F4/F5 key to switch. Required Maintenances are enabled and cannot be disabled. However, the maintenances that are meant to be done once, such as 1st mechanical check, will be disabled by "DONE" operation, to be explained in later section.

If multiple groups exist, press F2 [GROUP] and the group number to change the group to setup.

On software versions 7DC3/10 and later, press F3 DETAIL key displays Maintenance Item Setup screen, where type of interval and notification details can be setup.



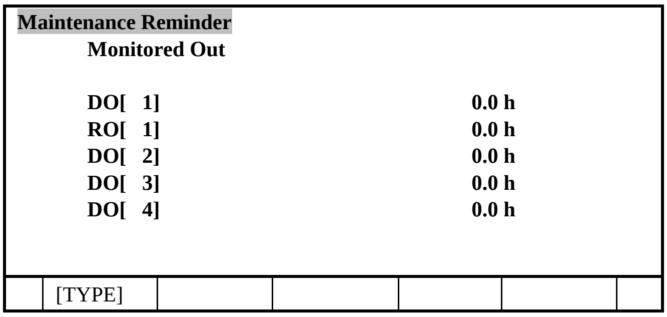
```
Maintenance Reminder
          Maintenance Item Setup                    1/12
          Group: 1
          Maintenance Name
            Days Elapsed:                  3   mo.    ENB

            Robot Running Time          ****   h      DSB
            DO[***] ON Time             ****   h      DSB
            R[****] value               ****          DSB
            J[*] movement               ****   kRev   DSB

            Notification Signal                DO[****]
            Completion Signal                  DI  [****]

            Maintenance notice
                           Days in advance:   ****  day
                              running time:   ****  h
            Warning interval
                                              ****  day
                              running time:   ****  h

   [TYPE]           OUT LIST
```

Table 54.2.2(b) Maintenance Item setup

Item	Description
Group	The group number, of which the maintenance item is currently shown.
Name	The name of the maintenance item. The names of the Required Maintenances are fixed. The names of Supplemental Maintenances can be set freely, up to 21 letters.
Interval	The interval, with which the maintenance should be done. The intervals can be set for “Days Elapsed” and/or “Robot Running Time”, “DO/RO ON Time”, “Register value”, or “Joint movement”.
DO/RO ON Time	Count the time specified DO or RO is ON, and notify maintenance based on the time length. With the cursor over “DO” (“RO”) pressing F5 “RO” (F4 “DO”) key will change the type of out. Enter the number of out to monitor ON time inside [] , and length of interval to the right. Up to 5 different outs can be monitored.
Register value	Notify the maintenance based on increase of specified register. Enter the number of register inside [], and length of interval, increase of register to the right. At the completion process, described later, the register value itself will not be reset. The next interval starts from the value at the time, and the notification will be based on the difference the current value and that value.
Joint movement	Notify the maintenance based on the amount of motion of the specified axis. Enter the number of the axis inside [], and length of interval, amount of motion to the right. The unit of interval is thousand revolutions (kRev) for rotational axis, and km for linear axis.
ENB/DSB	Enable or disable the interval type of the maintenance item. Select the item and press F4/F5 key to switch. Required Maintenances are enabled and cannot be disabled. However, the maintenances that are meant to be done once, such as 1st mechanical check, will be disabled by “DONE” operation, to be explained in later section. “Robot Running Time”, “DO/RO ON Time”, “Register value”, and “Joint movement” can only be enabled when the other three are disabled.
Notification Signal	This item sets item specific DO to output when the item reaches notice time. Setting 0 disables item specific DO.
Completion Signal	“Completion of Maintenance”, described in later section, can be processed through DI input. To process completion with DI, Notification Signal, described above, needs to be set. Completion Signal needs to be input until the falling edge of the Notification Signal is detected. If Completion Signal is input when Notification Signal is ON, Notification Signal will be OFF after the process. If Completion Signal is input when Notification Signal is OFF, Notification Signal will be ON after the process, then OFF after the time set in “Minimal DO after DI reset”, described in 54.2.1.

Item	Description
Maintenance Notice	This item sets timings to post warnings and output DO. The timing for “Days Elapsed” and for the enabled type of interval can be set. When the remaining time or value to the next maintenance reaches the set time, warnings, DO, and Notification screen start to remind maintenances. If not set, the common settings described in 54.2.1 will be used.
Warning interval	Set interval to post warnings after the maintenance notice time, explained above, has been reached. If not set, the common settings described in 54.2.1 will be used.

Pressing F3 “OUT LIST” displays the list of DO/ROs, of which ON time is counted. This screen is same screen as the one displayed when “OUT LIST” is pressed in Main Menu.



```
Maintenance Reminder
          Monitored Out

          DO[   1]                                   0.0 h
          RO[   1]                                   0.0 h
          DO[   2]                                   0.0 h
          DO[   3]                                   0.0 h
          DO[   4]                                   0.0 h

     [TYPE]
```

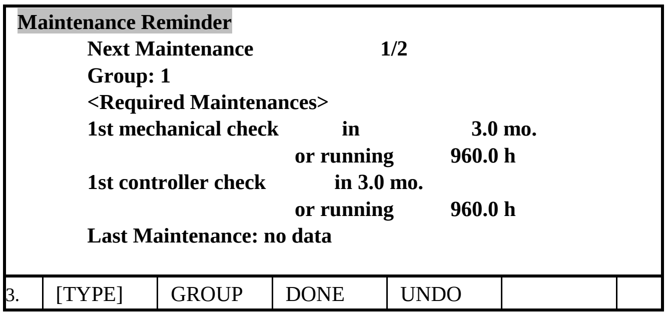
54.3 CHECK MAINTENANCE TIME AND COMPLETE MAINTENANCE

In the detail menu, remaining time to the next maintenance is displayed for each maintenance items. Also in this menu, an item can be “DONE” after the completion of the corresponding maintenance, to reset the counter to start counting for the next maintenance.

54.3.1 Check Maintenance Time

Detail menu is displayed with following procedures.

1. Display the main menu in the Section 54.1.
   This menu only displays the item of the group with the shortest remaining days in “Days Elapsed” and the item with the shortest remaining hours in “Robot Running Time”, if different from the item with the shortest remaining days.
2. Press F3 [DETAIL]



```
Maintenance Reminder
          Next Maintenance                 1/2
          Group: 1
          <Required Maintenances>
          1st mechanical check          in              3.0 mo.
                              or running          960.0 h
          1st controller check          in 3.0 mo.
                              or running          960.0 h
          Last Maintenance: no data

3.   [TYPE]     GROUP     DONE     UNDO
```

In detail menu, all the enabled items in the group are displayed with remaining time, beginning with Required Maintenances followed by Supplemental Maintenances. Upper lines represent the remaining time in “Days Elapsed” and the lower lines represent the remaining time in “Robot Running Time”. On software versions 7DC3/10 and later, the lower line shows “Robot Running Time”, “DO/RO ON Time”, “Register value”, or “Joint movement”. If only one type of the counter is enabled, the other type will display *’s.
If multiple groups exist, press F2 [GROUP] and the group number to change the group to show.

54.3.2 Maintenance Remind

When the maintenance time approaches, and remaining time/value is less than Maintenance notice configured in the Subsection 54.2.1, or 54.2.2 (on software versions 7DC3/10 and later) following warning is posted.
“DIAG-009 G(group number) (Name of the maintenance) in (remaining time) h”.
On 7DC3/10 and later:
“DIAG-009 G(group number) (Name of the maintenance) in (type of interval)(remaining time/value) (unit)”.
This warning is posted repeatedly with Warning Interval set in the section 54.2.1, or 54.2.2 (on 7DC3/10 and later) until the maintenance is processed “DONE” in the following section.
Also, if the output DO is set, the DO will be ON. This DO comes back ON even if switched OFF in I/O menu, unless the maintenance is processed “DONE.”
On software versions 7DC3/10 and later, Notification screen will notify the maintenance in addition to warnings and DO.

NOTE

This function only posts warning for the maintenance with the shortest remaining time per group. Even if several maintenances are within “Maintenance notice”, the warning for other maintenances will not be posted, unless the maintenance with the shortest remaining time is processed “done”.
On software versions 7DC3/10 and later, all maintenance items post warnings.

54.3.3 Upon Completion of Maintenance

When the Required Maintenances reach their maintenance time, follow the instructions in the mechanical unit operator’s manual to complete the corresponding maintenance, or contact your local FANUC representative. For Supplemental Maintenances, complete the corresponding maintenance in accordance with the design intent. Upon completion of the maintenance, “DONE” operation is necessary. If not, maintenance reminder warning will continue to be posted with set interval, and the next maintenance cannot be advised appropriately.

Follow the procedures to process the maintenance item as “DONE”

1. Display the detail menu in the Subsection 54.3.1.
2. Select the item of which the maintenance is completed.
3. Press F4 [DONE].
4. The prompt line will ask, “(Name of the maintenance) done?” Press F4 [yes].
   If wrong item is selected, press F5 [no].

The selected item is processed as “DONE” and the count to the next maintenance starts.
For the item that requires maintenance only once, such as “1st mechanical check,” the count will not restart, but displays “done”.

If an item is processed “done” by mistake, without completing the maintenance, it can be canceled for once. Undo the “done” process with following procedures.

1. Display the detail menu in the Subsection 54.3.1.
2. Select the item, which is processed “done” by mistake.
3. Press F4 [UNDO].
4. The prompt line will ask, “undo the last maintenance?” Press F4 [yes].
   If wrong item is selected, press F5 [no].
5. The remaining time changes back to as if it was not processed “done”.

After processing the item as “done,” selecting the item in the detail menu displays the date of “done” at the bottom of the screen after “Last Maintenance:”. If the item is never “done” or “done” is canceled with UNDO, it displays “no data”. When it displays “no data”, it cannot be undone.

On software versions 7DC3/10 and later, the completion of the maintenance can be processed by Completion Signal DI input. When the DI set in 54.2.2 is input, the item will be processed done. DI needs to be input until the process is confirmed. If the Notification Signal DO is ON, it will be OFF when the process is finished. If it is OFF, it will be ON when the process is finished, then OFF after “Minimal DO after DI reset” time set in 54.2.1. In either case, the process can be confirmed by detecting falling edge of Notification Signal DO.

Fig. 54.3.3 shows the flow of the system user should set up to utilize the completion process by DI input. If maintenance is done after the notification, with DO ON, DI should be input until DO turns OFF. If maintenance is done before the notification, with DO OFF, DI should be input until DO turns ON, then OFF. The DO would be ON at least for the time set in ”Minimal DO after DI reset”. It should be long enough so that the user system can detect DO ON. Since the signal for stopping DI input is always the DO change from ON to OFF, the user system for completion process can be set up as inputting DI until detecting falling edge of corresponding DO, regardless of whether DO was ON at the maintenance.

If Completion Signal DI is input by mistake, it needs to be “UNDONE” by operating the TP as described above.

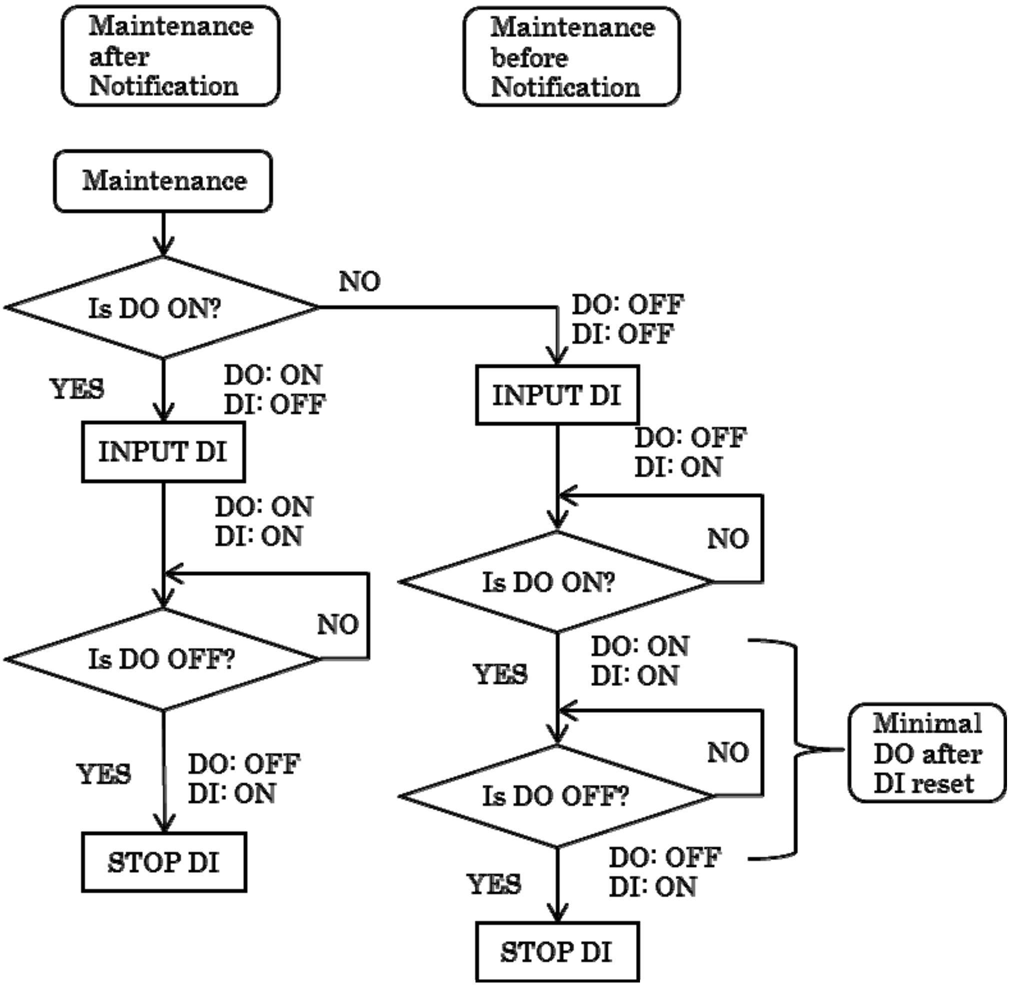


Fig. 54.3.3 The user system to utilize completion process by DI input

54.4 *i*RConnect

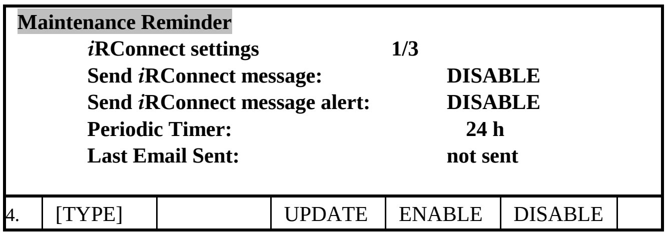
*i*RConnect is a function that sends email with information on the controller, which can be viewed with *i*RConnect mobile application. Using in conjunction with *i*RConnect, maintenance time can be checked on mobile devices. By sending periodic and reminder emails to the configured address, it enables checks on mobile devices. Please refer to the chapter on *i*RConnect from Ethernet Function Operator’s Manual for the use of *i*RConnect, such as setup. *i*RConnect requires software option, *i*RConnect (R818).

54.4.1 Setup for Maintenance Reminder

To check Maintenance Reminder information on iRConnect, in addition to setup on iRConnect such as server or address, some setup is required on Maintenance Reminder side.

Display iRConnect settings menu with following procedures.

1. Display the main menu in the Section 54.1.
2. Press F5 [iRConn]
   When iRConnect option is not ordered, [iRConn] key will not appear.



```
Maintenance Reminder
          iRConnect settings              1/3
          Send iRConnect message:                DISABLE
          Send iRConnect message alert:          DISABLE
          Periodic Timer:                          24 h
          Last Email Sent:                       not sent

4. | [TYPE] |        | UPDATE | ENABLE | DISABLE |
```

Table 54.4.1 Maintenance Reminder setup items for iRConnect

Item	Description
Send iRConnect message	Enable/Disable iRConnect message of Maintenance Reminder information. When All Emails in SETUP iRConnect menu is disabled, it cannot be enabled.
Send iRConnect message alert	Enable/Disable message to “Alert Addr” (SMS) when Maintenance Reminder warning is posted. When “Send iRConnect message” is disabled, it cannot be enabled.
Periodic Timer	Set interval with which iRConnect message is to be sent to “To Addr” when Send iRConnect message is enabled.
Last Email Sent	It displays the last time Maintenance Reminder created iRConnect message. If it is different from the time stamp on Maintenance Reminder screen on iRConnect application, it indicates email was not sent/received properly.

This setup is valid for all groups and maintenance items.

If for some reason, such as trouble on servers, email is not sent/received properly and information on iRConnect is not updated (when time stamp on iRConnect and “Last Email Sent” mentioned above do not correspond), new email can be created/sent at any time by pressing F3 [UPDATE].

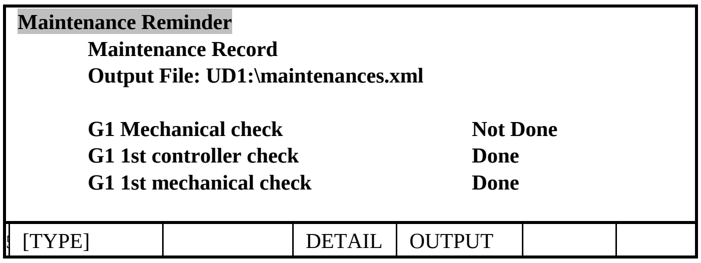
54.5 MAINTENANCE RECORD

On software versions 7DC3/10 and later, maintenance records are saved. The record is created at the time of first warning, and updated when past due and when completed. 10 last records can be viewed on TP. By creating maintenance record file beforehand, old records can be saved and viewed as xml file.

54.5.1 Display Maintenance Record

Display Maintenance Record screen with following procedures.

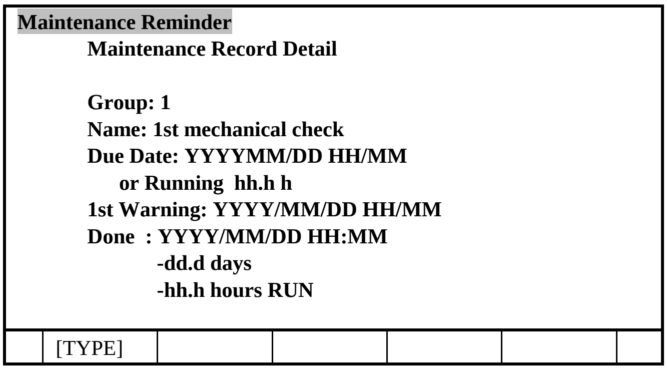
1. Display the main menu in the Section 54.1.
2. Press [NEXT] key.
3. Press F4 RECORD



```
Maintenance Reminder
          Maintenance Record
          Output File: UD1:\maintenances.xml

          G1 Mechanical check                 Not Done
          G1 1st controller check             Done
          G1 1st mechanical check             Done

 [TYPE]   |          |  DETAIL  |  OUTPUT  |       |
```

Pressing F3 DETAIL shows detail information of the item selected.

Pressing F4 OUTPUT creates the file specified and writes the information of the items that are “Done”. Once the file is created, the information of the items “Done” thereafter are added to the file.



```
Maintenance Reminder
          Maintenance Record Detail

          Group: 1
          Name: 1st mechanical check
          Due Date: YYYYMM/DD HH/MM
               or Running  hh.h h
          1st Warning: YYYY/MM/DD HH/MM
          Done  : YYYY/MM/DD HH:MM
                    -dd.d days
                    -hh.h hours RUN

     [TYPE]   |         |         |         |         |
```

Table 54.5.1 Maintenance Record items

Item	Description
Group	Group number of the item displayed.
Name	Name of the item displayed.
Due Date	Due date of the item displayed. The first line shows the due date calculated from the interval set in “Days Elapsed”. If not set, *’s will be displayed. The second line shows remaining time/value of the other type of interval. The negative value shows time/value past the due. If not set, the second line will not be shown.
1st Warning	It displays the date the first warning of the item is posted. If no warning is posted before processed “DONE”, *’s will be displayed.
Done	It displays the date the item is processed “DONE”. The second and third line shows the time/value past due at the completion process. The negative value shows remaining time/value when the item is processed “DONE” before due.

54.5.2 Maintenance Record File

Maintenance Record File is created in xml format at specified directory with specified name.

Table 54.5.2 Items in Maintenance Record File

Node/Element	Description
MAINT_RECORD	Maintenance Record Data
CONT	Controller
ID	ID set in 54.2.1. If not set, it would be other type of ID that is automatically set.
ROBOT	The group of the Maintenance Record Data
Model	The name of the model of the group. Ex. R-2000iC/165F
Group	The group number
ITEM	Maintenance Item
ID	The ID number of the maintenance item. It corresponds to the order the item is shown in the screen described in 54.2.2.
Name	The name of the maintenance item

Node/Element	Description
Type	The type of the maintenance item expressed in numbers. 1: “Days Elapsed” only. 2: “Robot Running Time” is set. 3: “DO/RO ON Time” is set. 4: “Register value” is set. 5: “Axis motion” is set.
HIST	The record of maintenance
DueAct	Due date and time in terms of “Days Elapsed”
DueOther	Due in other types of timer. The negative value means time/value till the due. The positive value means time/value past due. If the item is “done”, it is same as DonePast.
DueType	The type of the timer that reached due, expressed in number. 0:Not due. 1: “Days Elapsed”. 2: “Robot Running Time”. 3: “DO/RO ON Time”. 4: “Register value”. 5: “Axis motion”.
Warn	The date and time of the first warning.
DoneDate	The date and time when it is “done”.
DonePast	The difference between due and the timer when it is “done”. It is same as DueOther.

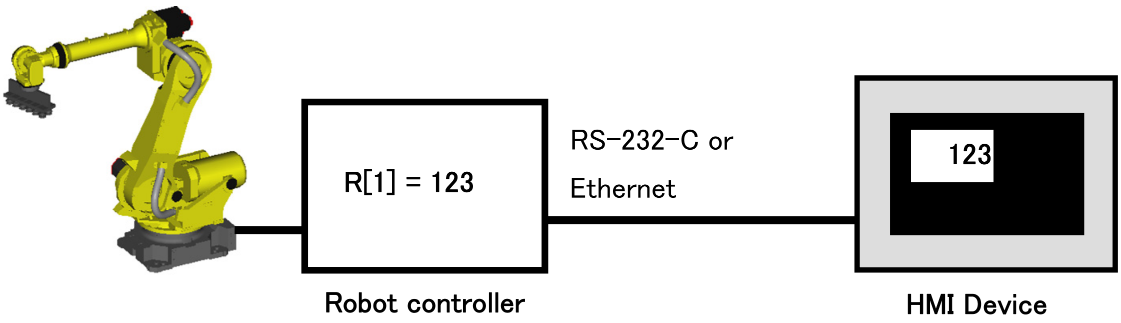
Example:

```
<?xml version="1.0"?>
<MAINT_RECORD>
<CONT ID="TEST ">
 <ROBOT Group="1" Model="R-2000iC/165F">
  <ITEM ID="1" Name=" 1st mechanical check" Type="2">
   <HIST DueAct="2016/3/16 16:08" DueOther="-3839.8 hours RUN" DueType="0" Warn="" DoneDate="2015/03/18 15:44:38" DonePast="-3839.8 hours RUN"/>
  </ITEM>
  <ITEM ID="11" Name="TEST1" Type="5">
   <HIST DueAct="" DueOther=" -100.0 kRev J[1]" DueType="0" Warn="2015/03/18 15:43:50" DoneDate="2015/03/18 15:44:06" DonePast=" -100.0 kRev J[1]"/>
  </ITEM>
 </ROBOT>
</CONT>
</MAINT_RECORD>
```

55 HMI DEVICE COMMUNICATION

HMI device communication function allows to communicate with external HMI device that is connected via RS-232-C or Ethernet. Various robot data, such as DI, DO, Register and so on, can be read or written from the HMI device by using this function.
This function works as MODBUS slave, and communicate with the HMI device that is MODBUS master.

HMI DEVICE can design the screen by putting lumps, switches and so on. The assignment of robot data to the lumps, switches and so on can be defined by user. For example, if robot register is assigned to the display panel on HMI device, the register value can be displayed on the screen of HMI device.



When "Standard setting (R651)" is selected, this function is standard. When "North America Setting (R650)" is selected, "HMI device (R553)" option is needed to use this function.

⚠ **WARNING**

This function checks the received data and performs the proper process only when the robot controller receives the data sent from HMI device. This function cannot detect the situation that the communication from HMI device is stopped accidentally. Therefore, this function must not be used to transfer the safety data. For example, when HMI device sends STOP request to robot, there is no guarantee that the robot is stopped by the request. It is very dangerous to make safety system by using this function.

NOTE

When this function is used to communicate with HMI device, please test the whole robot system including the HMI device.

Available HMI devices

Digital Electronics Corporation GP4000 Series

RS-232-C :	"Manufacturer: Modbus-IDA", "Series: MODBUS RTU SIO master"
Ethernet:	"Manufacturer: Modbus-IDA", " Series: MODBUS TCP master"

KEYENCE CORPORATION VT3 Series

RS-232-C:	"Manufacturer: MODBUS protocol", "Model: RTU mode (1:N)"
Ethernet:	"Manufacturer: MODBUS protocol", "Model: MODBUS/TCP(Ethernet)"

55.1 CONNECTION OF HMI DEVICE

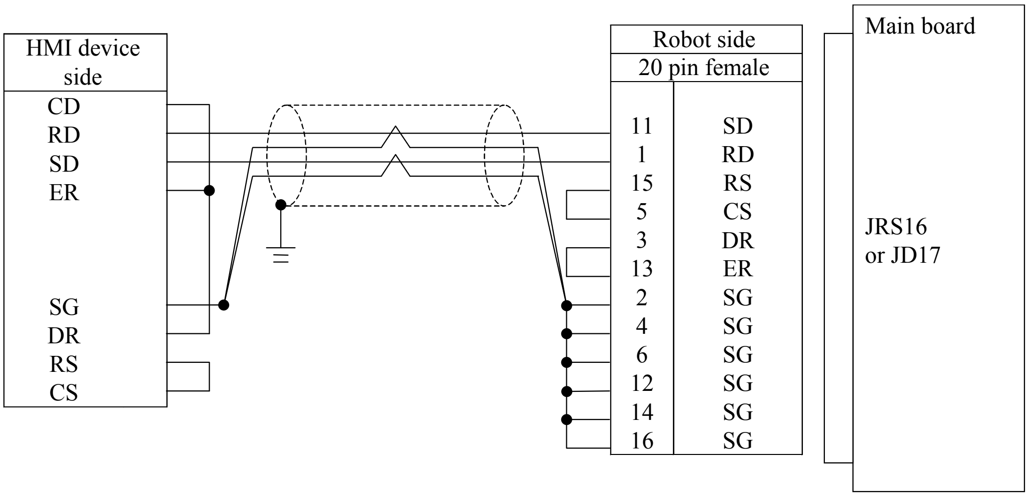
55.1.1 RS-232-C Connection

Cable

The following diagram shows the connection of RS-232-C. Please refer to the manual of the HMI device about the connection of HMI device side.

A-cabinet

The cable of the ROBOT CONTROLLER side is connected to JRS16 or JD17 connector on the main board.

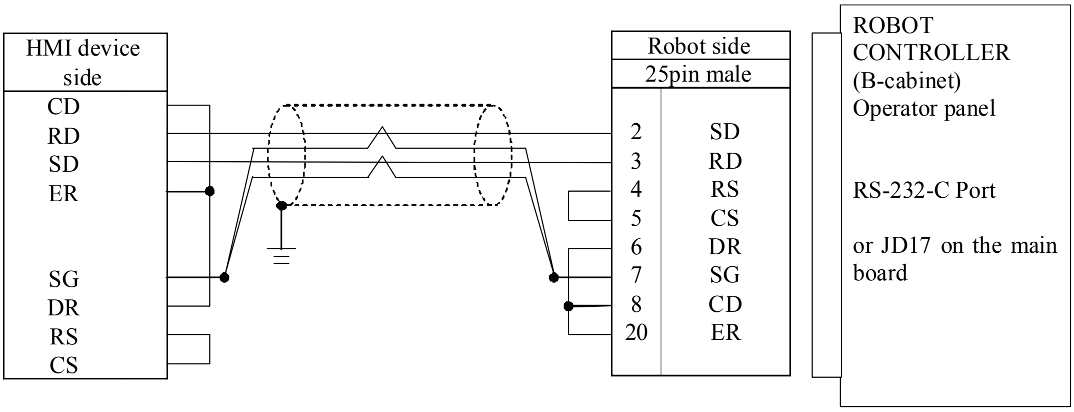


Specification of the connectors and the cable are as following:

- Connector PCR-E20FS (Honda Tsushin Kogyo)
- Housing PCR-V20LA (Honda Tsushin Kogyo), or compatible connector
- Shielding quality Entirely shielded twisted pair cable is recommended. For protection against the noise, the twisted pair wire should be used for pairs of RD and SG, SD and SG.

B-cabinet

The D-Sub 25pin is connected to the RS-232-C port in front side of ROBOT CONTROLLER operator panel or JD17 connector on the main board.



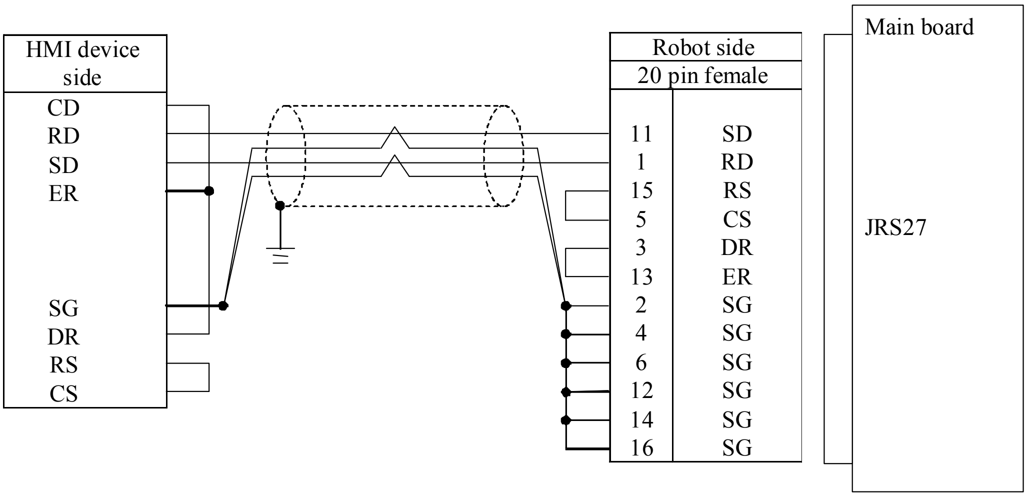
Specification of the connectors and the cable are as following (Operator panel):

- Connector D-Sub25pin male DBM-25P (ANSI/EIA-232)
- Housing DB-C2-J9 (ANSI/EIA-232)
- Shielding quality Entirely shielded twisted pair cable is recommended.

For protection against the noise, the twisted pair wire should be used for pairs of RD and SG, SD and SG.

R-30iB Mate

The cable of the ROBOT CONTROLLER side is connected to JRS27 connector on the main board.



Specification of the connectors are as following:

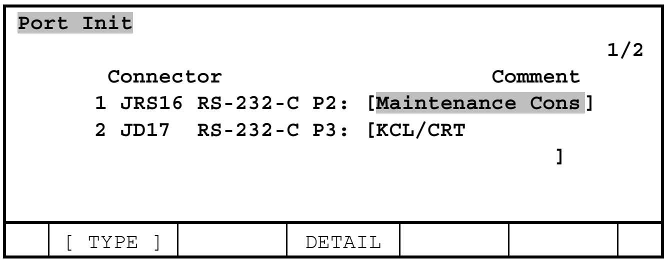
- Connector PCR-E20FS (Honda Tsushin Kogyo)
- Housing PCR-V20LA (Honda Tsushin Kogyo), or compatible connector
- Shielding quality Entirely shielded twisted pair cable is recommended.
  For protection against the noise, the twisted pair wire should be used for pairs of RD and SG, SD and SG.

Setting of Serial Port

To use HMI device communication via RS-232-C, it is necessary to set "HMI dev(MODBUS)" in Port Init menu on Robot controller.

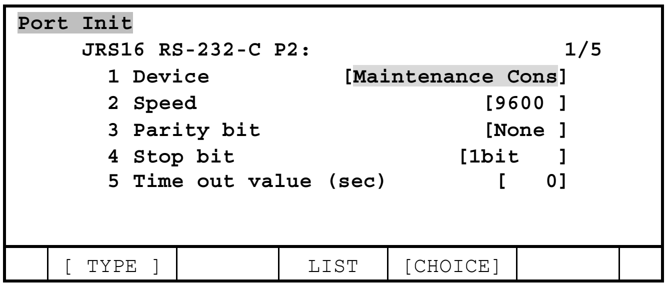
1 Press [MENU] key on teach pendant, and select "6 SETUP".

2 Press F1[TYPE] key, and select "Port Init".



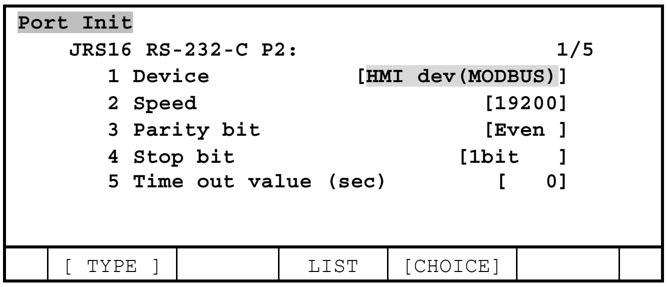
```
Port Init
                                                1/2
       Connector                      Comment
      1 JRS16  RS-232-C P2:  [Maintenance Cons]
      2 JD17   RS-232-C P3:  [KCL/CRT
                                             ]

  [ TYPE ]          DETAIL
```

3 Set cursor on "JRS16 RS-232-C P2:" line, and press F3(DETAIL) key. The detail menu of "JRS16 RS-232-C P2:" is displayed.



```
Port Init
     JRS16 RS-232-C P2:                     1/5
       1 Device          [Maintenance Cons]
       2 Speed                      [9600 ]
       3 Parity bit                 [None ]
       4 Stop bit                 [1bit   ]
       5 Time out value (sec)        [   0]

  [ TYPE ]          LIST     [CHOICE]
```

4 Set cursor on "Device" line, and press F4[CHOICE], and select "HMI dev(MODBUS)" from the list. There are "HMI device" and "HMI dev(MODBUS)" in the list. It is necessary to set "HMI dev(MODBUS)" for MODBUS communication. It is not possible to use "HMI device" and "HMI dev(MODBUS)" at the same time.



```
Port Init
    JRS16 RS-232-C P2:                        1/5
        1 Device            [HMI dev(MODBUS)]
        2 Speed                       [19200]
        3 Parity bit                  [Even ]
        4 Stop bit                   [1bit   ]
        5 Time out value (sec)         [   0]

   [ TYPE ]            LIST    [CHOICE]
```

5 Setup "Speed", "Parity bit" and "Stop bit" as same with the HMI device.

Setup slave address

The slave address of this function is set in the system variable $SNPX_PARAM.$MODBUS_ADR. The default value is 1. The slave address of MODBUS RTU can be changed by this system variable. This function responds to the request for the specified slave address only. The request for the slave address 0 is processed as broadcast.

55.1.2 Ethernet Connection

Setting of TCP/IP

To use HMI device communication via Ethernet, it is necessary to setup TCP/IP on Robot controller in advance.
Please refer to "SETTING UP TCP/IP" section in "Ethernet Function OPERATOR'S MANUAL".

Setting of Ethernet connection

Ethernet connection of HMI device is disable by default. To use Ethernet connection, please set the number of HMI devices that are connected at the same time to the system variable $SNPX_PARAM.$NUM_MODBUS (Default is 0). When Ethernet connection is requested from HMI devices more than the setting value, the HMI device whose latest communication is the oldest is disconnected automatically.

Unit ID

Robot controller does not use Unit ID in MODBUS TCP communication data. Robot controller responds all requests from HMI device regardless of Unit ID.

Port number

The port number of this function is set in the system variable $SNPX_PARAM.$MODBUS_PORT. The default value is 502. The port number of this function can be changed by this system variable.

MODBUS TCP Server (R800)

MODBUS TCP Server (R800) option is independent from this function. But when this option is loaded, if $SNPX_PARAM.$NUM_MODBUS is set to 1 or greater value, "PRIO-090 SNPX communication error", "HRTL-048 Address already in use" occurs and Ethernet communication of this function does not work. It is because this function and MODBUS TCP Server use the same port number (502).
By changing the port number of this function, both this function and MODBUS TCP Server can be used at the same time.

55.2 MODBUS COMMUNICATION

55.2.1 MODBUS data model

MODBUS bases its data model on a series of tables.

Table	Object type	Type of
Discrete input	Single bit	READ only
Coils	Single bit	READ-WRITE
Input Registers	16-bit word	READ only
Holding Registers	16-bit word	READ-WRITE

55.2.2 Correspondence of MODBUS Address to Robot Data

Each “Table” has address area 1 ~ 65535. This function defines the following correspondence of MODBUS address to Robot data. HMI device can read or write the corresponded robot data by accessing MODBUS address.

Table	Address range	Corresponded robot data (a: address)	
Discrete input	1 ~ 10000	Digital input	DI[a]
	10001 ~ 20000	Robot input	RI[a-10000]
	20001 ~ 21000	UOP input	UI[a-20000]
	21001 ~ 21999	UOP output	UO[a-21000]
	22000 ~ 21999	SOP input	SI[a-22000]
	23000 ~ 24000	SOP output	SO[a-23000]
	24001 ~ 25000	Weld interface digital input	WI[a-24000]
	25001 ~ 26000	Weld interface digital output	WO[a-25000]
	26001 ~ 27000	Wire soldering detector input	WSI[a-26000]
	27001 ~ 28000	Wire soldering detector output	WSO[a-27000]
	28001 ~ 65536	Not used	
Coils	1 ~ 10000	Digital output	DO[a]
	10001 ~ 20000	Robot output	RO[a-10000]
	20001 ~ 30000	Flag	F[a-20000]
	30001 ~ 65536	Not used	
Input Registers	1 ~ 1000	Group input	GI[a]
	1001 ~ 2000	Group output	GO[a-1000]
	2001 ~ 3000	Analog input	AI[a-2000]
	3001 ~ 4000	Analog output	AO[a-3000]
	4001 ~ 65536	Not used	
Holding Registers	1 ~ 16384	Robot data is assigned (assigned to R[a] by default)	
	16385 ~ 65536	Not used	

When the address that is not used or the corresponded robot data is not assigned is read from HMI device, the read value is always 0. When the address is written from HMI device, the request is ignored. In these cases, this function does not return error to the HMI device.

The address range 1 ~ 16384 of Holding Registers can be assigned to various Robot data. (→ Assignment of Holding Registers)

HMI device cannot write to the signals corresponded to Discrete input and Input Registers. It is possible to write these signals via Holding Registers by assigning these signals to Holding Registers. (→ Assign I/O data and simulation status)

In this function, address range is specified as 1 ~ 65536. But some HMI devices specify as 0000 ~ FFFF (start from 0 and specified as hexadecimal). In this case, please convert the specified address as subtraction of 1 and change to hexadecimal.

55.2.3 MODBUS Function Code

This function supports the following function code.

Function code name	Code
Read Coils	01h
Read Discrete Inputs	02h
Read Holding Registers	03h
Read Input Register	04h
Write Single Coil	05h
Write Single Register	06h
Write Multiple Coils	0Fh
Write Multiple Registers	10h

55.3 ASSIGNMENT OF HOLDING REGISTERS

The address range 1 ~ 16384 of Holding Registers can be assigned to various Robot data. Each address has 2 bytes (16 bits) memory space. It is possible to assign 32 bit data to the continuous 2 addresses, and to assign string data to the continuous plural addresses.

The following robot data can be assigned to the Holding Registers.

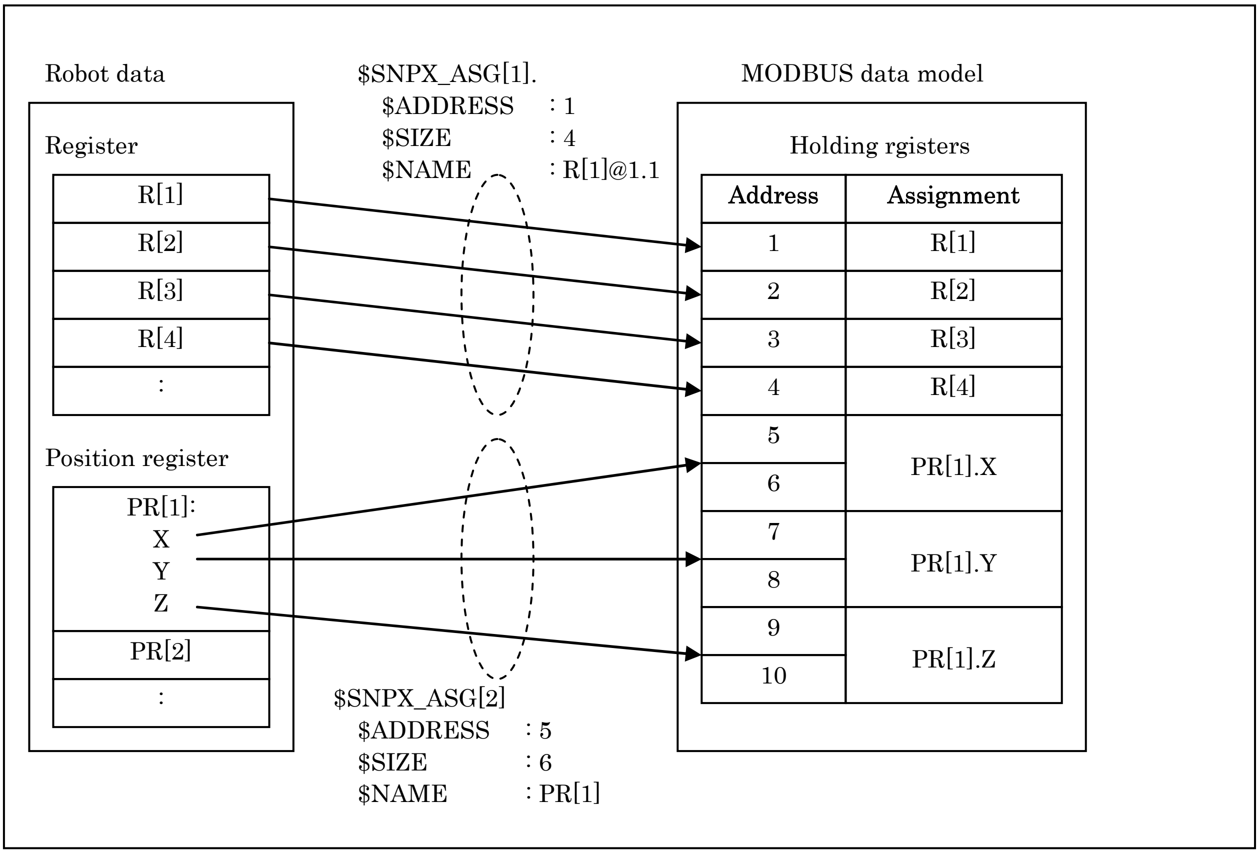
- Register data
- Position resister data
- String register data
- Current position
- Alarm history
- Program execution status (Running program name and line number)
- System variable data
- Comment of Register, Position register, String register and I/O
- I/O data and simulation status
- Integrated PMC address data
- Symbol and comment of Integrated PMC address.

The assignment between Robot data and Holding Registers is defined by setting of $SNPX_ASG. The $SNPX_ASG is 80 arrays of $SNPX_ASG[1]~[80], every element has the following variables. Various robot data can be assigned to the Holding Registers by setting $SNPX_ASG.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. Range: 1~16384
$VAR_NAME	Meaning: String to specify robot data. It specifies the robot data type and index by string. The meaning of this variable is different according to the assigned data type. Please refer to the description in each robot data. Example: R[1] : Register R[1] PR[1] : Position register PR[1] POS[1] : Current position of group 1 robot

Variable in $SNPX_ASG	Description
$MULTIPLY	Meaning: Multiply for assigned data. It specifies the conversion type from assigned data to Holding Registers. The meaning of this variable is different according to the assigned data type. Please refer to the description in each robot data. Example: When register is assigned and the register value is 123.45, the value of the Holding Registers is the following. When $MULTIPLY is 1, Holding Register is 123. When $MULTIPLY is10, Holding Register is 1235 When $MULTIPLY is 0.1, Holding Register is 12

The following figure shows a example that robot register R[1~4] are assigned to Holding Register address 1 ~ 4 and X,Y,Z of robot position register PR[1] are assigned to Holding Register address 5 ~ 10.



The default setting of $SNPX_ASG is the following.

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

By the default setting, the assignment of Holding Registers is the following.

Robot data	Address of Holding Registers	Example
Register R[*x*]	*x*	R[1] ⇔ Address 1

Note: Data type is 16 bits signed integer. Value is rounded off to no decimal place. Range of value is from -32768 to 32767. If value is out of the range, lower 16bits are accessed.

55.3.1 Data type of Holding Registers

Data type of Holding Register is 16 bits integer. Data range of Holding Register is -32768 ~ 32767. The value of assigned robot data is converted as follows according to the data type of the assigned data.

Assigned data	Description
16 bits signed integer Range: -32768 ~ 32767	The Holding Register data is the same as the assigned data.
32 bits signed integer Range: -2147483648 ~ 2147483647	The continued 2 Holding Registers express a 32 bits signed integer. The former address is lower 16 bits data, and the latter address is upper 16bits data.
Real	The continued 2 Holding Registers express a single precision binary real format of IEEE 754. The former address is lower 16 bits data, and the latter address is upper 16bits data. In case of that real type robot data is assigned to Holding Registers, when $MULTIPLY is 0, the robot data is assigned to Holding Registers as real type data. When $MULTIPLY is not 0, the robot data is assigned to Holding Registers as 32 bits signed integer. In this case, the Holding Registers has the value that the robot data is multiplied by $MULTIPLY and rounded off to no decimal place.
String	The continued Holding Registers express a string data. One Holding Register expresses 2 characters. Byte order of string data is specified by $MULTIPLY as follows. $MULTIPLY = 1 (0 or more): Lower byte is the first character, upper byte is the second character. $MULTIPLY = -1 (less than 0): Upper byte is the first character, lower byte is the second character. The last of the string data must be 0. When string data is written to Holding Registers, it is necessary to write 0 to the last of the string data.

⚠ **WARNING**

If Integer data is written to the Holding Registers of Real data type, the assigned robot data may be set to the value that is too big or cannot be processed. And it may cause serious effect such as the robot moves to unexpected position because the assigned position register has abnormal position data. Please set $MULTIPLY to the value of not 0 normally, and assign 32 bit signed integer data to Holding Registers.

55.3.2 Assign Robot Registers

$SNPX_ASG setting to assign Robot Registers to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. One Register uses two Holding Registers. (You can change the number of Holding Registers for one robot register by adding “@” in $VAR_NAME) Range: 1~16384

Variable in $SNPX_ASG	Description
$VAR_NAME	Meaning: String to specify robot data. Please set ”R[1]” to assign Register R[1]. The number in [ ] is the index of register. Continued registers such as R[2]-R[5] can be assigned by one $SNPX_ASG element. In this case, please set $SIZE to 8 because the number of registers to assign is 4. And please set $VAR_NAME “R[2]”. The index 2 means the starting index of the continuous assignment. If “@1.1” is added just after the string, the number of Holding Registers for one register is changed from 2 to 1. In this case, accessing data size becomes 16 bits. Example: R[1]@1.1 “R[1]” part means to assign robot registers from index 1. “@1.1” part means one register is accessed as 16 bits data.
$MULTIPLY	Meaning: The value accessed by HMI device is multiplied by $MULTIPLY. When $MULTIPLY is 0, it means that HMI can access Holding Registers as 32bits REAL data. When it is not 0, HMI accesses Holding Registers as 32bits signed integer. And value is rounded off to no decimal place. Range: 0.0001~10000, 0 Example: Register value is 123.45. When $MULTIPLY is 1, Holding Register is 123 When $MULTIPLY is 10, Holding Register is 1235. When $MULTIPLY is 0.1, Holding Register is 12. When $MULTIPLY is 0, Holding Register is 123.45 of REAL

For example, $SNPX_ASG is set as follows.

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

In this case, address of Holding Registers are corresponded to robot registers as follows.

Address	Assigned data
1	R[1] as 16bits signed integer.
2	R[2] as 16bits signed integer.
3-4	R[1] multiplied by 100 as 32bits signed integer.
5-6	R[2] multiplied by 100 as 32bits signed integer.
7-8	R[1] divided by 10 as 32bits signed integer.
9-10	R[2] divided by 10 as 32bits signed integer.
11-12	R[1] as 32bits REAL

$SNPX_ASG[1] defines that 2 Holding Registers from address 1 to 2 are assigned to robot registers from R[1] multiplied by 1 as 16 bits signed integer. One Holding Register address is 16 bits data, and one register uses one Holding Register. So, address 1 is assigned to R[1] as 16 bits signed integer and address 2 is assigned to R[2] as 16bits signed integer.

$SNPX_ASG[2] defines that 4 Holding Registers from address 3 to 6 are assigned to robot registers from R[1] multiplied by 100 as 32 bits signed integer. One robot register uses 2 Holding Registers. So, addresses 3-4 are assigned to R[1] multiplied by 100 as 32 bits signed integer and address 5-6 are assigned to R[2] multiplied by 100 as 32 bits signed integer.

$SNPX_ASG[3] defines that 4 Holding Registers from 7 to 10 are assigned to robot registers from R[2] divided by 10 as 32 bits signed integer. One robot register uses 2 Holding Registers. So, addresses 7-8 are assigned to R[2] divided by 10 as 32 bits signed integer and address 9-10 are assigned to R[3] divided by 10 as 32 bits signed integer.

$SNPX_ASG[4] defines that 2 Holding Registers from 11 to 12 are assigned to robot register R[1] as 32bit REAL. So, addresses 11-12 are assigned to R[1] as 32bits REAL

55.3.3 Assign Position Registers

$SNPX_ASG setting to assign Position Registers to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. One Position Register uses 50 Holding Registers. (You can change the number of Holding Registers for one position register by adding “@” in $VAR_NAME) Range: 1~16384
$VAR_NAME	Meaning: String to specify robot data. Please set ”PR[1]” to assign Position Register PR[1]. The number in [ ] is the index of position register. Continued position registers like PR[2]-PR[5] can be assigned by one $SNPX_ASG element. In this case, please set $SIZE 200 because the number of position registers to assign is 4. And please set $VAR_NAME “PR[2]”. The index 2 means the starting index of the continuous assignment. In multi group system, PR[1] means group 1 data of PR[1]. Please set PR[G2:1] to assign group 2 data of PR[1]. If “@” is added just after the string, the specified part of the position register is assigned. This is explained later.
$MULTIPLY	Meaning: The value accessed by HMI is multiplied by $MULTIPLY. Only REAL values are effected by this setting. When $MULTIPLY is 0, it means special that HMI can access Holding Registers as 32bits REAL data. When it is not 0, HMI accesses Holding Registers as 32bits signed integer. And value is rounded off to no decimal place. Range: 0.0001~10000, 0 Example: Position register member value is 123.45. When $MULTIPLY is 1, Holding Register is 123 When $MULTIPLY is 10, Holding Register is 1235. When $MULTIPLY is 0.1, Holding Register is 12. When $MULTIPLY is 0, Holding Register is 123.45 of REAL

⚠ **WARNING**

If Integer data is written to the Holding Registers of Real data type, the assigned robot data may be set to the value that is too big or cannot be processed. And it may cause serious effect such as the robot moves to unexpected position because the assigned position register has abnormal position data. Please set $MULTIPLY to the value of not 0 normally, and assign 32 bit signed integer data to Holding Registers.

NOTE

When the position register is locked by LOCK PREG instruction, the writing to the position register by this function is ignored. To confirm if the writing is succeeded or not, please read the position data just after the writing and confirm the read data.

One position register uses 50 Holding Registers. Contents of the 50 Holding Registers are the following.

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

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

- Position data has uninitialized data (displayed as "******" on Teach Pendant).
- Position data is Joint representation and it can not be converted to Cartesian.

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

Note2: VALIDJ shows whether the position data is valid for Joint or not. This becomes 0 at the following situation, and it becomes 1 in the other situation.

- Position data has uninitialized data (displayed "******" on Teach Pendant).
- Position data is Cartesian representation and it can not be converted to Joint.

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

Note3: UF is user frame number.

If UF is 0, world frame is used.

If UF is -1, the current selected user fame is used.

UF of Position Register is always -1.
This value can not be changed.

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

- Position register has 2 representations, Cartesian or Joint. If detail of the position register is displayed as X,Y,Z,W,P,R, it is Cartesian representation. If it is displayed as J1-6, it is Joint representation.
- HMI device can always access to any member without regarding to current representation. Position representation is changed by reading from HMI. But if the representation of position resister and the representation of accessed member from HMI are different, please note the following.
- If the position data is out of stroke limit or there is uninitialized member, this position data cannot be converted to another representation. In this case, the all members in the part of another representation becomes 0.
  For example, a position register is Cartesian representation and X is 10000, it is out of stroke limit, J1-J9 is read as 0 by HMI device. In this case, the members of Cartesian part or UF/UT part can be read correctly.
- If the representation of position resister and the representation of accessed member from HMI device are different, communication response time is increased. Because position representation conversion takes much time. If HMI device reads many position registers, this conversion time may make big effect to response time.

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

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

Extract a part by adding "@" in $VAR_NAME

If "@" is added to $VAR_NAME, the specified part of data structure is extracted. It was already used for Register assignment (R[1]@1.1).

For example, you need to access only X, Y and Z of PR[1-3]. In this case, normally the setting of $SNPX_ASG is the following.

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

The following is the correspondence between position registers and Holding Registers of this setting.

Address	Assigned data
1-2	X of PR[1] as 32bits signed integer
3-4	Y of PR[1] as 32bits signed integer
5-6	Z of PR[1] as 32bits signed integer
51-52	X of PR[2] as 32bits signed integer
53-54	Y of PR[2] as 32bits signed integer
55-56	Z of PR[2] as 32bits signed integer
101-102	X of PR[3] as 32bits signed integer
103-104	Y of PR[3] as 32bits signed integer
105-106	Z of PR[3] as 32bits signed integer

Actual read data are only 18 Holding Registers, but this assignment occupies 150 Holding Registers. And, some HMI accesses address 7-50 of Holding Registers that is not necessary to read, because reading one big data is more efficient than reading several small data from the communication point of view. But the address 27-45 includes Joint representation part, and if position data is Cartesian representation, representation conversion is needed to read this unnecessary data.

To communicate efficiently, please set $SNPX_ASG as follows.

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

The following is the correspondence between position registers and Holding Registers of this setting.

Address	Accessed data
1-2	X of PR[1] as 32bits signed integer
3-4	Y of PR[1] as 32bits signed integer
5-6	Z of PR[1] as 32bits signed integer
7-8	X of PR[2] as 32bits signed integer
9-10	Y of PR[2] as 32bits signed integer
11-12	Z of PR[2] as 32bits signed integer
13-14	X of PR[3] as 32bits signed integer
15-16	Y of PR[3] as 32bits signed integer
17-18	Z of PR[3] as 32bits signed integer

The change of $SNPX_ASG is that “@1.6” is added in $VAR_NAME and $SIZE is changed from 150 to 18. By this change, the number of Holding Registers for one position register is changed from 50 to 6. The “6” in ”@1.6” means the number of Holding Registers for one position register. And the “1” in “@1.6” means the starting address of extracting part in position data structure.
By specifying “@” in $VAR_NAME, you can extract the specified part of position data structure.

@1.6

The size of extracting part (The number of Holding Registers for one position)

The starting address of extracting part in position data structure.

You can specify “@” not only for position register, but also for all data that is assigned by $SNPX_ASG.

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

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

55.3.4 Assign String Registers

$SNPX_ASG setting to assign String Registers to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. One String Register uses 40 Holding Registers. It is possible to deal with first 80 characters of one string register in default setting. (You can change the number of Holding Registers for one string register by adding “@” in $VAR_NAME) Range: 1~16384

Variable in $SNPX_ASG	Description
$VAR_NAME	Meaning: String to specify robot data. Please set ”SR[1]” to assign String Register SR[1]. The number in [ ] is the index of string register. Continued string registers such as SR[2]-SR[5] can be assigned by one $SNPX_ASG element. In this case, please set $SIZE 160 because the number of string registers to assign is 4. And please set $VAR_NAME “SR[2]”. The index 2 means the starting index of the continuous assignment. If “@” is added just after the string, the specified part of the String Register is assigned. Example: SR[1]@1.50 “SR[1]” means to assign String Registers from index 1. “@1.50” means the first 100 characters of String Register data are assigned.
$MULTIPLY	Meaning: Specify byte order of string. Range: 1,-1

55.3.5 Assign Current Position

$SNPX_ASG setting to assign Current Position to Holding Registers is the following.

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

Variable in $SNPX_ASG	Description
$MULTIPLY	Meaning: The assigned data is multiplied by $MULTIPLY. Only REAL values are effected by this setting. If $MULTIPLY is 0, it means that the data is assigned to Holding Registers as 32bits REAL data. If it is not 0, it means that the data is assigned to Holding Registers as 32bits signed integer. And value is rounded off to no decimal place. Range: 0.0001~10000, 0 Example: Current position member value is 123.45. When $MULTIPLY is 1, Holding Register is 123 When $MULTIPLY is 10, Holding Register is 1235. When $MULTIPLY is 0.1, Holding Register is 12. When $MULTIPLY is 0, Holding Register is 123.45 of REAL

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

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

Current position is not assigned continuously. For example, the following setting of $SNPX_ASG assigns current position of user frame 1 to address 1-50, but current position of user frame 2 is not assigned to address 51-100. Please set “POS[2]” in another $SNPX_ASG.

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

55.3.6 Assign Alarm History

$SNPX_ASG setting to assign Alarm history to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. One alarm uses 100 Holding Registers. (You can change the number of Holding Registers for one alarm by adding “@” in $VAR_NAME) Range: 1~16384
$VAR_NAME	Meaning: String to specify robot data. Please set ”ALM[1]” to assign alarm. The number in [ ] is line number in alarm menu. The latest alarm is 1. “ALM[1]” assigns active alarm. Alarms displayed in active alarm menu are assigned. “ALM[E1]” assigns alarm history. Alarms displayed in alarm history menu are assigned. “ALM[M1]” assigns motion alarm history. Alarm displayed in motion alarm menu are assigned. “ALM[S1]” assigns system alarm history. Alarm displayed in system alarm menu are assigned. “ALM[A1]” assigns application alarm history. Alarm displayed in application alarm menu are assigned. “ALM[P1]” assigns password log. Password log displayed in password log menu are assigned.

Variable in $SNPX_ASG	Description
$MULTIPLY	Meaning: Specify byte order of string. Range: 1,-1

Note: Alarm history is read only. If you write it, nothing occurs.

One alarm uses 100 Holding Registers. Contents of the 100 Holding Registers are the following.

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

Facility code and alarm number of "RESET" is read as 0. Alarm message is "RESET".

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

In string item, addresses after string are 0.

One alarm uses 100 Holding Registers, if you would like to read only facility code and alarm number, almost of 100 Holding Registers area is not used. To communicate efficiently, please use "@".

Example: $SNPX_ASG is set as follows.

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

The correspondence of address and alarm data of this setting is the following.

Address	Assigned data
1	Alarm ID of alarm1
2	Alarm number of alarm1
3	Alarm ID of cause code of alarm1
4	Alarm number of cause code of alarm1
5	Alarm ID of alarm2
6	Alarm number of alarm2
7	Alarm ID of cause code of alarm2
8	Alarm number of cause code of alarm2
9	Alarm ID of alarm3
10	Alarm number of alarm3
11	Alarm ID of cause code of alarm3
12	Alarm number of cause code of alarm3

55.3.7 Assign Program Execution Status

$SNPX_ASG setting to assign program execution status to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. One task uses 18 Holding Registers. (You can change the number of Holding Registers for one task by adding “@” in $VAR_NAME) Range: 1~16384
$VAR_NAME	Meaning: String to specify robot data. Please set ”PRG[1]” to assign program execution status. The number in [ ] is task number. In single task system, program execution status always can be assigned by PRG[1]. In multi task system, when 2 tasks are running at the same time, PRG[1] and PRG[2] shows the execution status of each task. Which task is PRG[1] is decided by timing of execution and communication. But the task that is read as PRG[1] is always read as PRG[1] until it is aborted.
$MULTIPLY	Meaning: Specify byte order of string. Range: 1,-1

Note: Program execution status is read only. If you write it, nothing occurs.

One task uses 18 Holding Registers. Contents of the 18 Holding Registers are the following.

Address	Description
1-8	Program name 16 characters string Name of running program. When sub program is called, it shows sub program name.
9	Line number 16bits signed integer. Execution line number. When sub program is called, it shows line number of sub program.
10	Execution status 16bits signed integer Aborted 0 Paused 1 Running 2

Address	Description
11-18	Parent program name 16 characters string Name of the started program When sub program is not called, it is the same as program name.

When program is aborted, all members become 0.

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

$VAR_NAME	Program name and line number
PRG[1]	Program name Name of running program. Line number Execution line number.
PRG[M1]	Program name When running program is MACRO program, it shows name of parent program that calls the running program. When the parent program is also MACRO program, it shows name of parent program that is not MACRO program. Line number Line number of CALL instruction that calls the running program. When all parent programs are MACRO program. It becomes 0.
PRG[K1]	Program name When running program is KAREL program, it shows name of parent program that calls the running program. When the parent program is also KAREL program, it shows name of parent program that is not KAREL program. Line number Line number of CALL instruction that calls the running program. When all parent programs are KAREL program. It becomes 0.
PRG[MK1] or PRG[KM1]	Program name When running program is MACRO or KAREL program, it shows name of parent program that calls the running program. When the parent program is also MACRO or KAREL program, it shows name of parent program that is not MACRO nor KAREL program. Line number Line number of CALL instruction that calls the running program. When all parent programs are MACRO or KAREL program. It becomes 0.

By adding “@”, a part of this structure is assigned to Holding Registers.
Example: $SNPX_ASG is set as follows.

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

The correspondence of address and program execution status of this setting is the following.

Address	Accessed data
1	Line number of task 1
2	Execution status of task 1
3	Line number of task 2.
4	Execution status of task 2.

55.3.8 Assign System Variables

$SNPX_ASG setting to assign system variables to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. The number of Holding Registers for one system variable is defined by data type of the system variable. Please refer the following data type list. (You can change the number of Holding Registers for one system variable by adding "@" in $VAR_NAME) Range: 1~16384
$VAR_NAME	Meaning: String to specify robot data. Please set system variable name, for example "$SNPX_ASG[1].$ADDRESS". If array system variable such as "$SNPX_ASG[1]" is set, array elements are assigned continuously like Registers. KAREL string can be specified by the following format. $[KAREL program name]variable name.
$MULTIPLY	Meaning: The meaning of $MULTIPLY is defined according to the data type of system variable. Please refer the following data type list.

To access to system variable, data type of assigned Holding Registers and meaning of $MULTIPLY are changed according to data type of system variable. The following is the list of data type of system variables that can be assigned to Holding Registers, and it also explain the number of Holding Registers and meaning of $MULTIPLY.

Data type of system variables	The number of Holding Registers for one system variable	Meaning of $MULTIPLY
INTEGER 32bits signed integer	2	The data is multiplied by $MULTIPLY. The data is accessed as 32 bits signed integer. And value is rounded off to no decimal place. If $MULTIPLY is 0, it is same as $MULTIPLY is 1.
SHORT 16bits signed integer	2	
BYTE 8bits signed integer	2	
REAL 32bits real	2	The data is multiplied by $MULTIPLY. If $MULTIPLY is 0, it means that the data is accessed as 32 bits REAL data. If it is not 0, the data is accessed as 32 bits signed integer. And value is rounded off to no decimal place
BOOLEAN TRUE/FALSE	2	$MULTIPLY is not used for this data type. The data is accessed as 32bit signed integer. If value is TRUE, Holding Register is 1. If it is 0, Holding Register is 0. If 0 is written, the system variable becomes FALSE. If not 0 is written, the system variable becomes TRUE.
POSITION Position data	50	Data structure is the same as position register. Meaning of $MULTIPLY is also same as position register.
STRING String data	40	Specify byte order of string.

INTEGER, SHORT and BYTE are accessed by the same way. The system variable that integer value is displayed in system variable menu on teach pendant is INTEGER, SHORT or BYTE.
The variable that real value is displayed in system variable menu on teach pendant is REAL.
The variable that TRUE or FALSE is displayed in system variable menu on teach pendant is BOOLEAN.
The variable that string is displayed on system variable menu on teach pendant is STRING.
The variable that is displayed as POSITION in system variable menu on teach pendant is POSITION.

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

⚠ **WARNING**
If Integer data is written to the Holding Registers of Real data type, the assigned robot data may be set to the value that is too big or cannot be processed. And it may cause serious effect such as the robot moves to unexpected position because the assigned POSITION type variable has abnormal position data. Please set $MULTIPLY to the value of not 0 normally, and assign 32 bit signed integer data to Holding Registers.

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

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

This assigns system variables to Holding Registers as follows.

Address	**Accessed data**
1-2	X of user frame 1 of Group 1 (X of $MNUFRAME[1,1])
3-4	Y of user frame 1 of Group 1 (Y of $MNUFRAME[1,1])
5-6	Z of user frame 1 of Group 1 (Z of $MNUFRAME[1,1])
7-8	X of user frame 2 of Group 1 (X of $MNUFRAME[1,2])
9-10	Y of user frame 2 of Group 1 (Y of $MNUFRAME[1,2])
11-12	Z of user frame 2 of Group 1 (Z of $MNUFRAME[1,2])

55.3.9 Assign comment of R[ ], PR[ ] , SR[ ] and I/O

$SNPX_ASG setting to assign comment of Register, Position register, String register and I/O to Holding Registers is the following.

Variable in $SNPX_ASG	**Description**
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1 to 16384
$SIZE	Meaning: Number of Holding Registers to assign. One comment uses 40 Holding Registers. (You can change the number of Holding Registers for one comment by adding “@” in $VAR_NAME) Range: 1 to 16384

Variable in $SNPX_ASG	Description
$VAR_NAME	Meaning: String to specify robot data. Please set "R[C1]" to assign comment of register. The number in [ ] is index of register. To assign comment of position register, string register and I/O, please set the following string. Register R[C1] Position register PR[C1] String register SR[C1] DI DI[C1] DO DO[C1] RI RI[C1] RO RO[C1] UI UI[C1] UO UO[C1] SI SI[C1] SO SO[C1] WI WI[C1] WO WO[C1] WSI WSI[C1] WSO WSO[C1] GI GI[C1] GO GO[C1] AI AI[C1] AO AO[C1] Comment of the continued registers such as R[2]~R[5] can be assigned by one $SNPX_ASG element. In this case, please set $SIZE 160 because the number for one comment to assign is 40. And please set $VAR_NAME "R[C2]". The index 2 means the starting index of the continuous assignment. If "@" is added just after the string, the specified part of the comment is assigned. Example: R[C1]@1.5 "R[C1]" means to assign comment of registers from index 1. "@1.5" means the first 10 characters of comment are assigned.
$MULTIPLY	Meaning: Specify byte order of string. Range: 1,-1

55.3.10 Assign I/O data and simulation status

$SNPX_ASG setting to assign I/O data and simulation status to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. The number of Holding Registers for one I/O value or simulation status is changed by $MULTIPLY setting. When $MULTIPLY is 1, one I/O value or simulation status uses one Holding Register. When $MULTIPLY is 0, one Holding Register includes 16 I/O value or simulation status as bit image. (Note: Value of GI/O and AI/O use one Holding Register even if $MULTIPLY is 0.) Range: 1~16384

Variable in $SNPX_ASG	Description
$VAR_NAME	Meaning: String to specify robot data. Please set ”DI[1]” to assign DI[1] data. The number in [ ] is index of DI. To assign I/O data or simulation status of the other type of I/O, please set the following string. Value Simulation DI DI[1] DI[S1] DO DO[1] DO[S1] RI RI[1] RI[S1] RO RO[1] RO[S1] UI UI[1] UO UO[1] SI SI[1] SO SO[1] WI WI[1] WI[S1] WO WO[1] WO[S1] WSI WSI[1] WSI[S1] WSO WSO[1] WSO[S1] GI GI[1] GI[S1] GO GO[1] GO[S1] AI AI[1] AI[S1] AO AO[1] AO[S1] Flag F[1] Marker M[1] Example: Assign DI[11]~DI[42]: When $MULTIPLY is 1, please set 32 to $SIZE because the number of DI is 32, and set “DI[11]” to $VAR_NAME. In this case, index 11 means that the start index of the assignment is 11. When $MULTIPLY is 0, 16 DI data are assigned to one Holding Register, so please set 2 to $SIZE, and set “DI[11]” to $VAR_NAME.
$MULTIPLY	Meaning: One Holding Register has one I/O data or 16 I/O data as bit image. One I/O data is assigned to one Holding Register1 16 I/O data are assigned to one Holding Register0

You can select by $MULTIPLY setting whether one I/O data is assigned to one Holding Register or 16 I/O data are assigned to one Holding Register.

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

When $MULTIPLY is 0, 16 I/O data or 16 simulation status are assigned to one Holding Register. When the value is ON, the corresponded bit of Holding Register is set to 1. When the value is OFF, the corresponded bit of Holding Register is set to 0. I/O of lower index is assigned to lower bit of Holding Register.

Example: When DI[1-16] is assigned to Holding Register, if only DI[1] is ON and the others are OFF, the Holding Register is 1. If only DI[16] is ON and the others are OFF, the Holding Register is 32768 (in case of unsigned 16 bit integer).

55.3.11 Assign Integrated PMC address data

$SNPX_ASG setting to Integrated PMC address data to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. The number of Holding Registers corresponded to PMC address is different by the specified PMC address that is byte address or bit address. When bit address is specified, one bit PMC address is assigned to one Holding Register. When byte address is specified, two bytes PMC addresses are assigned to one Holding Register. Range: 1~16384
$VAR_NAME	Meaning: String to specify robot data. Set "PMC1:X0" to assign data of PMC address X0. The "PMC1" part is PMC path number. "PMCS" means DCS Safety PMC. Data of bit address is assigned by specifying bit address such as "PMC1:X0.0". Example: Assign X0.0~X1.7 data of 1st path PMC When $SIZE is set to 1 and $VAR_NAME is set to "PMC1:X0", two byte address X0~X1 are assigned to one Holding Register address. Lower byte is assigned to X0, upper byte is assigned to X1. When $SIZE is set to 16 and $VAR_NAME is set to "PMC1:X0.0", 16 bit address X0.0~X1.7 are assigned to 16 Holding Registers. X0.0 is assigned to Address 1, X0.1 is assigned to Address 2, X1.7 is assigned to Address 16.
$MULTIPLY	Not used

NOTE

When K900.4 (Change address value) is 0, writing to PMC address data is ignored. To check if the writing is ignored or not, please read the PMC address after writing and confirm the read value.

Note: Address of DCS safety PMC is read only. If you write it, nothing occurs.

Example: $SNPX_ASG is set as follows.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	2	PMC1:R0.0	1
$SNPX_ASG[2]	3	2	PMC1:D0	1

The correspondence between Holding Registers and robot data by this setting is the following.
Bit address is assigned to the address 1 - 2, therefore 1 bit of PMC address is assigned to 1 Holding Register continuously.
Byte address is assigned to the address 3 - 4, therefore 2 bytes of PMC address is assigned to 1 Holding Register continuously.

Address	Assigned robot data
1	R0.0
2	R0.1
3	D0 - D1 (Lower byte:D0, Upper byte:D1)
4	D2 - D3 (Lower byte:D2, Upper byte:D3)

55.3.12 Assign Symbol and Comment of Integrated PMC address

$SNPX_ASG setting to symbol and comment of Integrated PMC address to Holding Registers is the following.

Variable in $SNPX_ASG	Description
$ADDRESS	Meaning: Start address of Holding Registers to assign. Range: 1~16384
$SIZE	Meaning: Number of Holding Registers to assign. One symbol or comment uses 40 Holding Registers. (You can change the number of Holding Registers for one symbol and comment by adding "@" in $VAR_NAME.) Range: 1~16384
$VAR_NAME	Meaning: String to specify robot data. Set "PMC1:S:X0" to assign symbol of PMC address X0. Set "PMC1:C:X0" to assign comment of the PMC address. The "PMC1" part is PMC path number. "PMCS" means DCS Safety PMC. Symbol and comment of bit address is assigned by specifying bit address such as "PMC1:S:X0.0". Example: PMC1:S:X0.0@1.5 The "PMC1:S:X0.0" part means that symbol of bit address from X0.0 is assigned continuously. The @1.5 part means that the top 10 characters are assigned.
$MULTIPLY	Meaning: Specify byte order of string. Range: 1,-1

Note: Symbol and comment of Integrated PMC are read only. If you write it, nothing occurs.

Example: $SNPX_ASG is set as follows.

	$ADDRESS	$SIZE	$VAR_NAME	$MULTIPLY
$SNPX_ASG[1]	1	10	PMC1:S:R0.0@1.5	1
$SNPX_ASG[2]	11	10	PMC1:C:D0@1.5	1

The correspondence between Holding Registers and robot data by this setting is the following.

Address	Assigned robot data
1-5	Symbol of R0.0
6-10	Symbol of R0.1
11-15	Comment of D0
16-20	Comment of D1

55.3.13 Hints

Assigned data is not read correctly

If HMI device reads Holding Register that is not assigned to any data, this Holding Register is always read as 0. If Holding Register that the problem occurs is not 0, this Holding Register is assigned to the other data. For example, when $SNPX_ASG is set as follows, addresses 101-150 of Holding Registers are assigned by both $SNPX_ASG[1] and $SNPX_ASG[2].

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

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

Therefore, addresses 101-150 are assigned to R[101]-R[150], and PR[1] can not be accessed by HMI device.
If the read data of Holding Register is 0, there may also be duplicated assignment as above. Please check duplication of $SNPX_ASG.
If $SNPX_ASG setting has problem even though assignment is not duplicated, please check $VAR_NAME of $SNPX_ASG.
The following list shows the correct format of $VAR_NAME setting. If the setting is not match to them, robot data is not assigned.

Correct format	Note
"R[n]"	
"PR[n]" "PR[Gn:n]	If you specify group number, ":" must be specified between group number and index.
"SR[n]"	
"POS[n]" "POS[Gn:n]	If you specify group number, ":" must be specified between group number and index.
"ALM[n]" "ALM[En]" "ALM[Pn]	Alarm line number must be specified just after E, M, A, S or P. The ":" must not be specified.
"PRG[n]" "PRG[Mn]", "PRG[Kn]" "PRG[MKn]", "PRG[KMn]"	Index must be specified just after M, K, MK or KM. The ":" must not be specified.
System variable name	The first character of system variable must be "$".
$[KAREL program name]variable name	Specify KAREL program name just after '$[', then specify ']' and specify KAREL variable name just after ']'.
"DI[n]", "DI[Sn]", "DI[Cn]" "DO[n]", "DO[Sn]", "DO[Cn]" "RI[n]", "RI[Sn]", "RI[Cn]", "RO[n]", "RO[Sn]", "RO[Cn]" "UI[n]", "UI[Cn]" "UO[n]", "UO[Cn]" "SI[n]", "SI[Cn]" "SO[n]", "SO[Cn]" "WI[n]", "WI[Sn]", "WI[Cn]", "WO[n]", "WO[Sn]", "WO[Cn]" "WSI[n]", "WSI[Sn]", "WSI[Cn]" "WSO[n]", "WSO[Sn]", "WSO[Cn]" "GI[n]", "GI[Sn]", "GI[Cn]" "GO[n]", "GO[Sn]", "GO[Cn]" "AI[n]", "AI[Sn]", "AI[Cn]" "AO[n]", "AO[Sn]", "AO[Cn]" "F[n]", "F[Cn]" "M[n]", "M[Cn]"	Index must be specified just after S or C. The ":" must not be specified.
PMC<path>:address PMC<path>:address.bit PMC<path>:S:address PMC<path>:S:address.bit PMC<path>:C:address PMC<path>:C:address.bit	The <path> is from 1 to 5 or "S". The ":" must be specified just after PMC path. And PMC address must be specified just after the ":". For symbol or comment, ":S" or ":C" must be specified just after PMC path.

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

Method to improve response time

- Position representation conversion takes much time. If you access position register, please assign only necessary members of position structure by using “@”. And please change position representation of the position registers to the representation that is accessed by HMI device.

56 FAULT & INCIDENT REPORT

FAULT & INCIDENT REPORT is a function to analyze alarms occurred in the past. It is possible to display the top5 faults about the number of incident, total time, and the longest time with graph.

This function is optional function. To use this function, “FAULT & INCIDENT REPORT option (R544) is required.

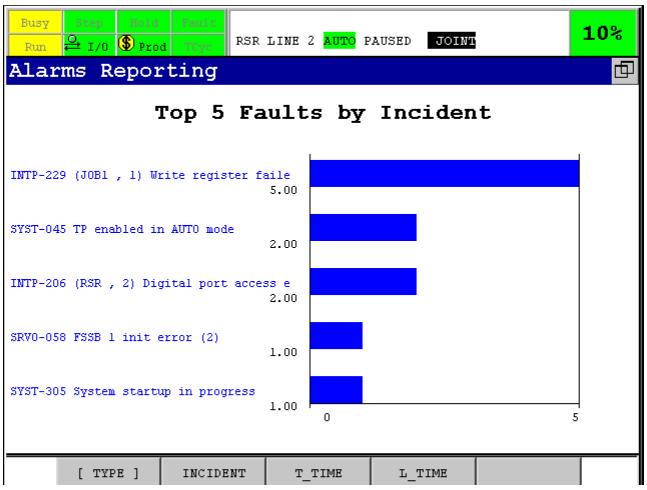
56.1 ALARMS REPORTING SCREEN

You can check the top 5 alarms about the number of incident, total time, and longest time of the alarms which are occurred at AUTO mode.
You can display Alarms Reporting Screen by selecting [MENU] key ->Alarm->F1[TYPE]->Reporting.

Alarms whose severity are NONE or WARN are not displayed on this screen.

INCIDENT

The top 5 faults by incident are displayed. Faults are located in order of the number occurred in the past. Each value and bar graph means the number of incident. This number is counted when the target fault is occurred at AUTO mode. (However, only alarms “SYST-038 Operation mode T1 Selected” and “SYST-039 Operation mode T2 Selected” are counted at T1mode, T2mode too)



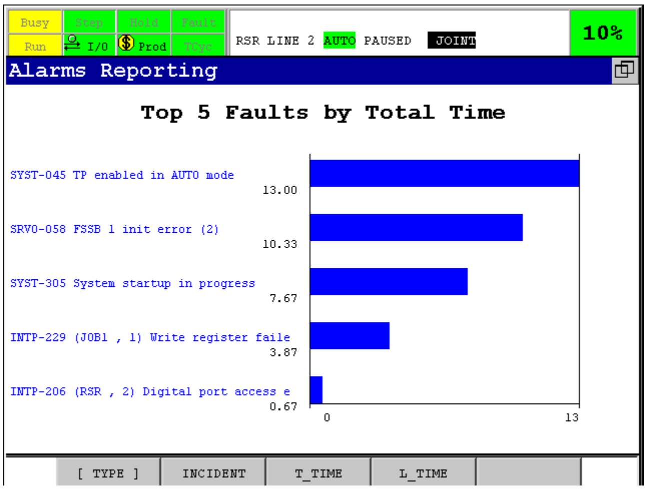
Top 5 Faults by Incident

```
INTP-229 (JOB1 , 1) Write register faile
                                    5.00
SYST-045 TP enabled in AUTO mode
                                    2.00
INTP-206 (RSR , 2) Digital port access e
                                    2.00
SRVO-058 FSSB 1 init error (2)
                                    1.00
SYST-305 System startup in progress
                                    1.00
```

Fig. 56.1(a) Alarms Reporting INCEIDENT Screen

Total Time(T_TIME)

The top 5 faults by Total Time are displayed. Faults are located in order of the total time occurred in the past. Each value and bar graph means the total time (unit is 1/10 minute). The incident time means time after the target fault is occurred at AUTO mode before you press the fault reset key. And the total time means sum of the incident times. (However, only alarms “SYST-038 Operation mode T1 Selected” and “SYST-039 Operation mode T2 Selected” are recorded at T1mode, T2mode too)



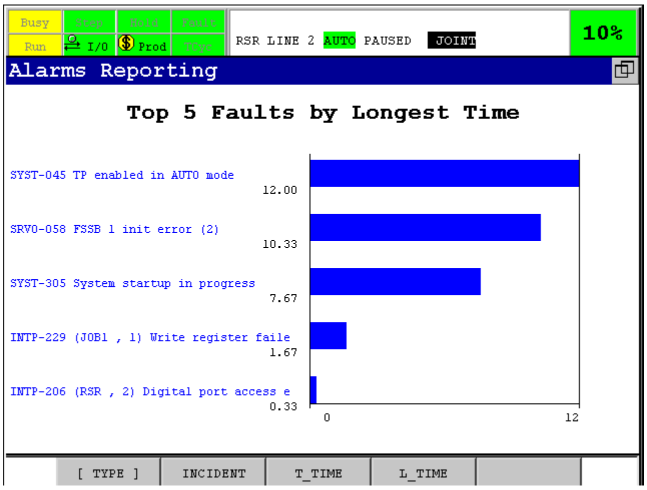
Top 5 Faults by Total Time

SYST-045 TP enabled in AUTO mode 13.00
SRVO-058 FSSB 1 init error (2) 10.33
SYST-305 System startup in progress 7.67
INTP-229 (JOB1 , 1) Write register faile 3.87
INTP-206 (RSR , 2) Digital port access e 0.67

Fig. 56.1(b) Alarms Reporting Total Time Screen

Largest Time(L_TIME)

The top 5 faults by Largest Time are displayed. Faults are located in order of the largest time occurred in the past. Each value and bar graph means the largest time (unit is 1/10 minute). The incident time means time after the target fault is occurred at AUTO mode before you press the fault reset key. And the largest time means the largest time in the incident times. (However, only alarms “SYST-038 Operation mode T1 Selected” and “SYST-039 Operation mode T2 Selected” are recorded at T1mode, T2mode too)



Top 5 Faults by Longest Time

SYST-045 TP enabled in AUTO mode 12.00
SRVO-058 FSSB 1 init error (2) 10.33
SYST-305 System startup in progress 7.67
INTP-229 (JOB1 , 1) Write register faile 1.67
INTP-206 (RSR , 2) Digital port access e 0.33

Fig. 56.1(c) Alarms Reporting Longest Time Screen

INDEX

REVISION RECORD

Edition	Date	Contents
05	Sep.,2015	• Following new functions have been added. VISUAL DIAGNOSTICS, PDF VIEWER FUNCTION, HELP AND DIAGNOSTICS DISPLAY, MAINTENANCE REMINDER, HMI DEVICE COMMUNICATION • Correction of errors
04	Aug.,2013	Addition of new software series 7DC2 (V8.20).
03	Mar.,2013	Addition of R-30*i*B Mate.
02	Nov.,2012	• Following new functions have been added. PAYLOAD OVER AND PAYLOAD SETTING CONFIRM FUNCTION, DATA TRANSFER BETWEEN ROBOTS FUNCTION, TOUCH SENSING, TOUCH SKIP FUNCTION, MROT INSTRUCTION, ROBOT ISOLATION FUNCTION • The function names have been modified in the following functions. SINGULARITY AVOIDANCE FUNCTION, CYCLE TIME LOGGING • The chapter of COLLISION DETECTION FOR AUXILIARY AXIS has been moved to "FANUC Robot series R-30*i*B CONTROLLER OPERATOR'S MANUAL (Basic Operation)" (B-83284EN). • Specifications have been modified in some functions. • Some errors in writing have been corrected.
01	Sep., 2012	

B-83284EN-2/05

* B - 8 3 2 8 4 E N - 2 / 0 5 *