

**FANUC Robotics
SYSTEM R-J2 Controller
Handling Tool Setup and
Operations Manual**

Version 4.40-1

MARO2HT4405801E REV. B

This publication contains proprietary information of FANUC Robotics North America, Inc. furnished for customer use only. No other uses are authorized without the express written permission of FANUC Robotics North America, Inc.

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

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

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

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



WARNING

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

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

FANUC Robotics North America, Inc.
Training Department
3900 W. Hamlin Road
Rochester Hills, Michigan 48309-3253
Tel: (248)377-7234
FAX: (248)377-7367 or (248)377-7362

**Copyright © 1998 by FANUC Robotics North America, Inc.
All Rights Reserved**

The information illustrated or contained herein is not to be reproduced, copied, translated into another language, or transmitted in whole or in part in any way without the prior written consent of FANUC Robotics North America, Inc.

AccuStat® , ArcTool® , DispenseTool® , FANUC LASER DRILL® , KAREL® , INSIGHT® , INSIGHT II® , PaintTool® , PaintWorks® , PalletTool® , SOCKETS® , SOFT PARTS® SpotTool® , TorchMate® , and YagTool® are Registered Trademarks of FANUC Robotics.

FANUC Robotics reserves all proprietary rights, including but not limited to trademark and trade name rights, in the following names:

AccuFlow®
ARC Mate®
ARC Mate Sr.®
IntelliTrak®
LaserTool®
MotionParts®
PaintWorks II®
PalletMate®
SureWeld®
TurboMove®

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

3,906,323	4,549,276	4,727,303	4,852,024	4,969,795
4,274,802	4,549,846	4,728,247	4,852,114	4,970,370
4,289,441	4,552,506	4,728,872	4,855,657	4,970,448
4,299,529	4,554,497	4,732,526	4,857,700	4,972,080
4,336,926	4,556,361	4,742,207	4,859,139	4,972,735
4,348,623	4,557,660	4,742,611	4,859,845	4,973,895
4,359,815	4,562,551	4,750,858	4,866,238	4,974,229
4,366,423	4,575,666	4,753,128	4,873,476	4,975,920
4,374,349	4,576,537	4,754,392	4,877,973	4,979,127
4,396,973	4,591,944	4,771,222	4,892,457	4,979,128
4,396,975	4,603,286	4,773,523	4,892,992	4,984,175
4,396,987	4,626,756	4,773,813	4,894,594	4,984,745
4,406,576	4,628,778	4,774,674	4,894,596	4,988,934
4,415,965	4,630,567	4,775,787	4,894,908	4,990,729
4,416,577	4,637,773	4,776,247	4,899,095	5,004,968
4,430,923	4,638,143	4,777,783	4,902,362	5,006,035
4,431,366	4,639,878	4,780,045	4,903,539	5,008,832
4,458,188	4,647,753	4,780,703	4,904,911	5,008,834
4,462,748	4,647,827	4,782,713	4,904,915	5,012,173
4,465,424	4,650,952	4,785,155	4,906,121	5,013,988
4,466,769	4,652,203	4,796,005	4,906,814	5,034,618
4,475,160	4,653,975	4,805,477	4,907,467	5,051,676
4,479,673	4,659,279	4,807,486	4,908,559	5,055,754
4,479,754	4,659,280	4,812,836	4,908,734	5,057,756
4,481,568	4,663,730	4,813,844	4,908,738	5,057,995
4,482,289	4,672,287	4,815,011	4,916,375	5,060,533
4,482,968	4,679,297	4,815,190	4,916,636	5,063,281
4,484,855	4,680,518	4,816,728	4,920,248	5,063,295
4,488,242	4,697,979	4,816,733	4,922,436	5,065,337
4,488,746	4,698,777	4,816,734	4,931,617	5,066,847
4,489,821	4,700,118	4,827,203	4,931,711	5,066,902
4,492,301	4,700,314	4,827,782	4,934,504	5,075,534
4,495,453	4,701,686	4,828,094	4,942,539	5,085,619
4,502,830	4,702,665	4,829,454	4,943,759	5,093,552
4,504,771	4,706,000	4,829,840	4,953,992	5,094,311
4,530,062	4,706,001	4,831,235	4,956,594	5,099,707
4,530,636	4,706,003	4,835,362	4,956,765	5,105,136
4,538,639	4,707,647	4,836,048	4,965,500	5,107,716
4,540,212	4,708,175	4,837,487	4,967,125	5,111,019
4,542,471	4,708,580	4,842,474	4,969,109	5,111,709
4,543,639	4,712,972	4,851,754	4,969,722	5,115,690
4,544,971	4,723,207			

FANUC Robotics – *Technical Support Hotline*

1-800-47-ROBOT

(1-800-477-6268)

Local/Internal 248-377-7159

Customer Service Center (Press 1)

Marketing and Sales Department (Press 2)

Technical Service (Press 1)

Tel (248) 377-7159

Fax: (248) 377-7463

24 Hour Hotline

- Technical Service Hotline support
- Service personnel dispatch
- After-hours parts support (8:00 pm to 8:00 am)

Information to have available

- Customer Number (if known)
- Company name
- Your name
- Your phone and fax numbers
- Robot and controller type
- “F#” or serial number of robot
- “Hour Meter” reading
- Software type and edition
- Any error messages and LED displays (if applicable)
- Your P.O. number for warranty, down robots, or preventive maintenance service orders

Parts (Press 2)

Tel (248) 377-7278

Fax: (248) 377-7832

8:00 am to 8:00 pm Monday to Friday

- Parts for down robots
- Replenishment part order
- Warranty part replacement
- Robot Software

Information to have available

- Customer Number (if known)
- Company name
- Your name
- Your phone and fax numbers
- Part name and number (if known)
- “F#” or serial number of robot
- “Hour Meter” reading
- Your P.O. number for warranty, down robots, and software orders
- Any error messages and LED displays (if applicable)

Training (Press 3)

Tel (248) 377-7234

Fax: (248) 377-7367

8:00 am to 5:00 pm Monday to Friday

- Training class registration
- Consultation for special training or on-site requests

Information to have available

- Customer Number (if known)
- Company name
- Your name
- Your phone and fax numbers
- Your shipping or billing address
- Types of courses needed
- Robot and controller type
- Number of people attending
- Method of payment

Part Repair (Press 4)

Tel (248) 377-7944

Fax: (248) 377-7367

8:00 am to 5:00 pm Monday to Friday

- Repair of electronic components
- Repair of mechanical components (wrists etc.)
- Warranty part repair

Information to have available

- Customer Number (if known)
- Company name
- Your name
- Your phone and fax numbers
- “F#” or serial number of robot
- “Hour Meter” reading
- Project number or P.O. number
- Shipping & billing addresses
- Reason for repair (any symptoms, error codes, or diagnostic LEDs that were identified)

*****NOTE: PLEASE OBTAIN A RETURN GOODS NUMBER (RGN) AUTHORIZATION FROM “PARTS” BEFORE SHIPPING ANY PARTS BACK TO OUR FACILITY. THE RGN IS NECESSARY FOR PROPER RECEIVING AND TRACKING.**

UPDATES FOR V4.40

This section lists the updates that have been made to the HandlingTool product for V4.40 in the following areas:

	Page
TCPMATE	-1
Overview	-1
Setting up TCPMate	-2
TCPMate Troubleshooting	-23
TCPMate Adjustment	-27
TCPMate Error Recovery	-31

.1 TCPMATE

.1.1 Overview

The TCPMate option provides a cost effective, easy-to-use solution to adjust the Tool Center Point (TCP) automatically. TCPMate automatically compensates for bent tools to reduce weld defects and increase system productivity.

When you use TCPMate, you start with a system that is fully functional and has a properly defined TCP. Then, you set up TCPMate and master the TCP. After it is mastered, TCPMate can be used at any time to check and adjust the TCP to compensate for problems such as a bent tool body. Since TCPMate is an executable program, it can be called either automatically (for example, every 50 production parts or between each cycle), or manually, whenever desired.

TCPMate compensates for variation of the TCP in the following directions:

- x and y
- x, y, and z
- z only

TCPMate can compensate for variation in orientation (w, p) but not for variation in r. However, most process applications are not significantly impacted if rotation about the tool z axis (r) varies slightly due to bent tooling.

TCPMate works with any robot and tool where the TCP is at the end of cylindrically shaped tooling. If the normal tooling is not cylindrical, or if the TCP is away from the tooling, a teach tip can be installed.

Note You should run TCPMate after every cap change procedure.

For ArcTool applications, TCPMate provides similar functionality to TorchMate (plus additional features), but does not require electrical continuity to determine the TCP correction.

To use TCPMate, you must do the following:

1. Install the TCPMate software option and any required hardware.
2. Set up TCPMate.
3. Include a CALL to TCPMate in your production teach pendant program (for automatic operation).
4. Run TCPMate.
5. Monitor TCPMate operations.

In order to use TCPMate successfully, review the following requirements and guidelines:

-
- TCPMate can be used with all end-of-arm safety devices and mounting arms, as long as they stay constant. TCPMate will recover the TCP of a tool or teach tip that can be inserted into a circular hole. TCPMate can be used with virtually any circular opening on a solid surface, such as a 6mm steel plate attached to cell fixturing. The circular opening should be approximately 10mm larger in diameter than the diameter of the largest expected recoverable bent tool. For example, if you suspect that the tool might bend 5mm and the tool shaft is 25mm in diameter, then the recommended circular opening would be 40mm. Larger openings are allowed, but the cycle time to complete a TCP recovery will increase proportionately.
 - In ArcTool, for tool orientation recovery with tooling such as wire feed torches, which have a very small TCP shank (weld wire) and a much larger supporting shaft (torch shaft), the two circle method should be used. For this case, the small circle should be no larger than the diameter of the larger supporting shaft. This circle is used for detecting the xy location of the weld wire. The z location is determined by moving the tooling into the small circle. If the small circle is larger than the supporting shaft, the z location cannot be determined. The larger circle should have a diameter as described above for the single circle case.
 - Compensation of the variation in z direction, and compensation of yaw and pitch (w, p) orientation axes, is optional.
 - While TCPMate works well even when the TOOL frame is inaccurate, the TOOL frame should be set.
 - The payload must be set accurately, especially when the TCPMate reference positions are at extreme points of the robot's reach.

After you have established the TCPMate TCP, you can use it in new and existing programs. You can apply TCPMate easily to new or existing programs without having to redefine positions as long as the original programs were created with an accurate tool center point.

.1.2 Setting Up TCPMate

You must install the TCPMate software option before you can use TCPMate. Refer to the *FANUC Robotics SYSTEM R-J3iB Software Installation Manual* for more information about installing software options.

Note During installation, TCPMate creates a program called TCPMATE. You can call this program whenever you require automatic operation. Refer to [Section .1.4](#) for more information.

Setting Up TCPMate

After you have installed the TCPMate software option, you must set up TCPMate as follows:

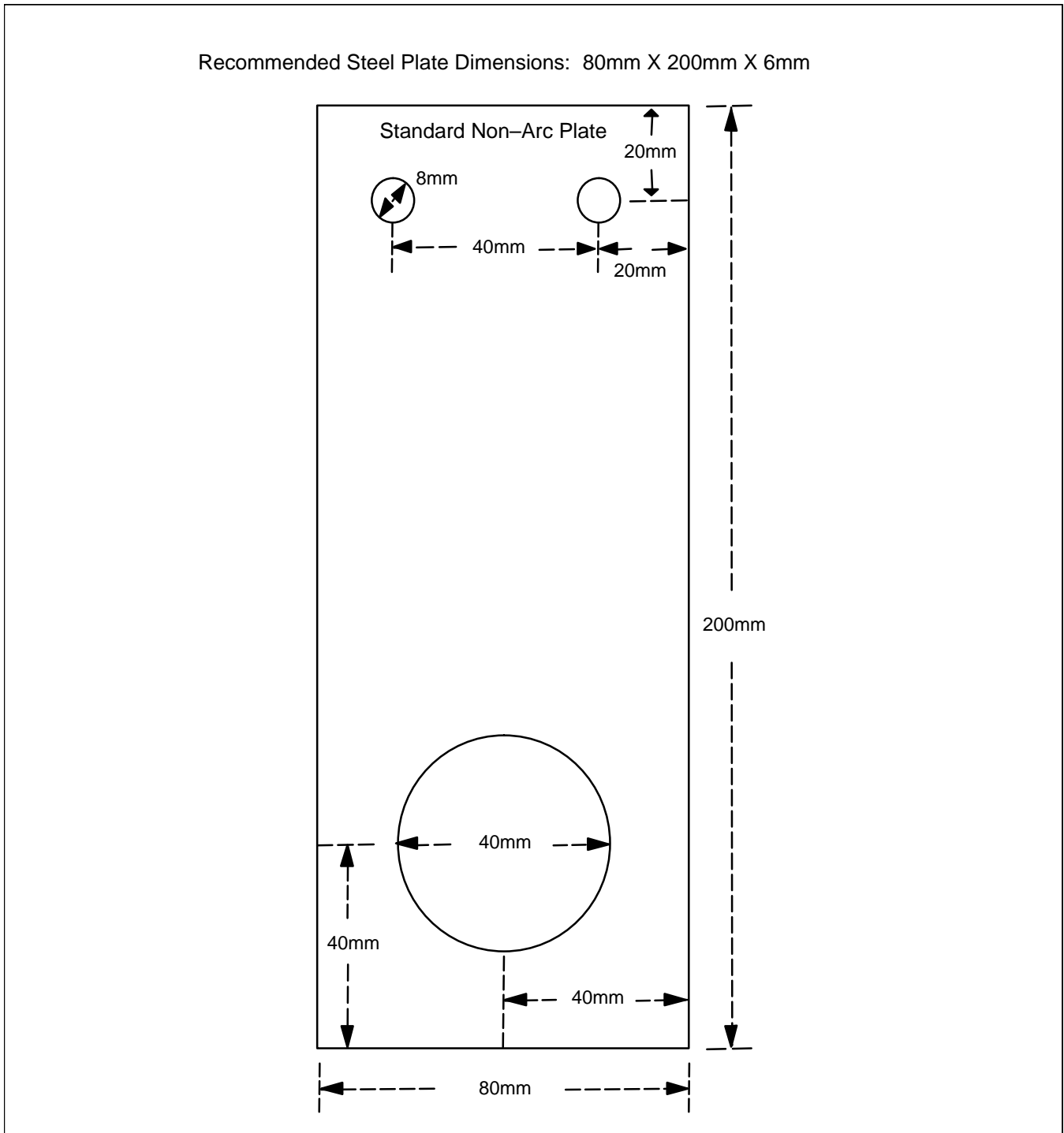
1. Install the touch plate on a stationary object in the robot workcell.
2. Select TCPMate check mode.
3. Set up the TCPMate option.

-
4. Master the TCP reference.

.1.2.1 Touch Plate Installation

Before you can define the TCPMate tool frame and master the TCP, you must install a touch plate on a stationary object within the robot workcell. See [Figure –1](#) for illustrations of the touch plate.

Figure -1. Recommended TCPMate Touch Plate Specifications



Use [Procedure -1](#) to install and align the touch plate.

Procedure -1 Installing and Aligning the Touch Plate

Conditions

- The TOOL frame has be set up correctly.

Refer to for more information.

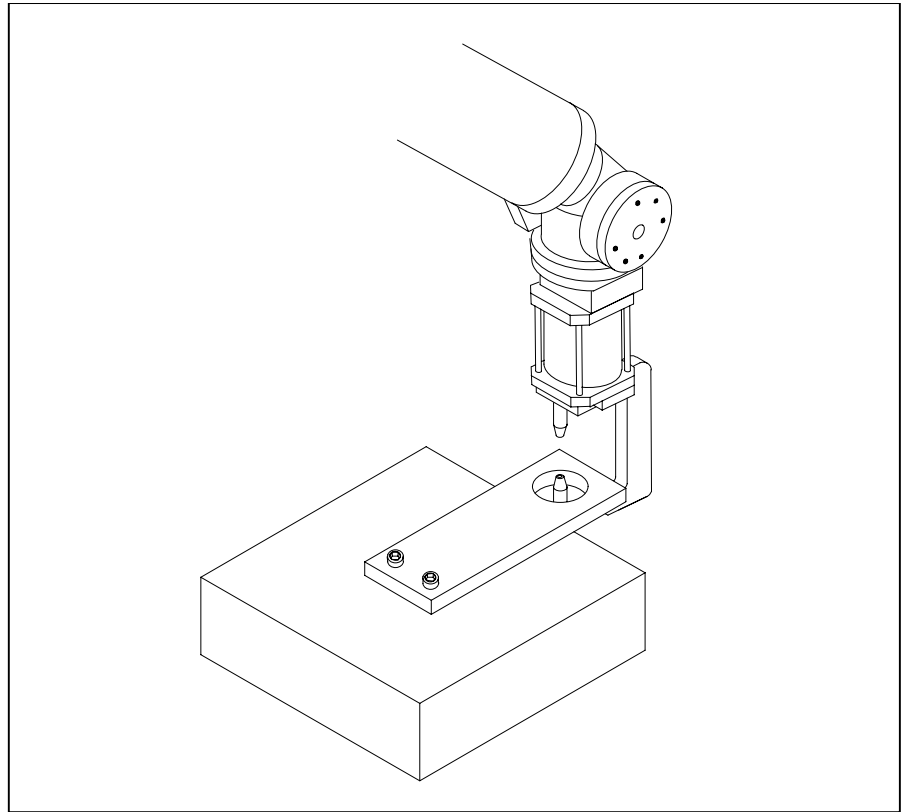
Steps

1. Install the touch plate on a flat level surface well within the workcell. See [Figure -2](#).

The touch plate should be securely fastened to a firm surface. If the touch plate position changes due to collisions or other events, the recovered TCP will be incorrect. There should be ample clearance below the plate so that the tooling can be inserted through the circular opening in the touch plate.

If you are using automatic TCPMate adjustment, place the plate in a location which minimizes motion from the last weld position used.

Figure –2. Installing the Touch Plate



2. Align the touch plate so that the plate is perpendicular to the specified robot WORLD axis. The default and typical frame is +z, meaning that the plate is in the WORLD x-y plane and motion from the reference position in the -z direction will cause the tool to go through the first circular hole.

In most applications, the typical TCP definition is that the TOOL z axis is along the shaft used by TCPMate. For arc welding applications, the TOOL +z direction extends from the TCP location into the torch body. For other applications, the TOOL +z direction extends away from the shaft at the TCP location. TCPMate works with any TOOL frame definition.

The typical location of the touch plate is parallel to the floor of the workcell and perpendicular to one of the robot axes.

- If the robot is floor mounted, the selected circle axis will be WORLD z.
- If the robot is invert mounted, the circle axis will be WORLD -z.
- If the robot is mounted at 90 degrees, the circle axis will be WORLD -x.

Installation and alignment of the touch plate is now complete.

Note If you are using ArcTool and require compatibility with TorchMate, refer to the “TorchMate” chapter in the *FANUC Robotics SYSTEM R-J3iB Controller ArcTool Setup and Operations Manual* for further information. TCPMate can coexist with TorchMate.

.1.2.2 TCPMate Mode Selection

Before you can use TCPMate, you must select the TCPMate mode. Refer to [Table –1](#) for listings and descriptions of the TCPMate mode setup items.

Use [Procedure -2](#) to set up the TCPMate mode.

Note If orientation recovery is used, it is important that the TCP location reflect the TCP that is along the shaft. If the TCP location is inaccurate, you might need to run TCPMate twice to recover the TCP accurately.

Table –1. TCPMate Mode Setup Items

ITEM	DESCRIPTION
Current Tool	This item is the UTOOL number and the current tool value.
TCP Offset Units: mm	This item is the difference between the current tool and the mastered tool.
Mode Values: TCP XY, TCP XYZ, TCP XYZWPR, TCP Z, QUICK XY, TCP ALIGN	This item is the type of motion and offset that TCPMate will correct.
Setup Values: COMPLETE or INCOMPLETE	This item indicates whether TCPMate setup is complete. If Setup is INCOMPLETE, you can not run TCPMate.

Table -1. TCPMate Mode Setup Items (Cont'd)

ITEM	DESCRIPTION
Reference Values: COMPLETE or INCOMPLETE	This item indicates whether the tool reference setup is complete. If Reference is INCOMPLETE, you can not run TCPMate.
Tool Log Values: ENABLED or DISABLED Default: ENABLED	This item indicates whether the TCP log utility is enabled. When ENABLED, the tool log stores the x, y, z, w, p, and r values for the previous tool.

Procedure -2 Selecting the TCPMate Mode

Steps

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select TCPMate. You will see a screen similar to the following.

TCPMate Menu		
Current Tool:	[1]	
X:	214.359	Y: -111.097
W:	89.338	P: 3.116
TCP Offset[mm]:	.088	
X:	.011	Y: -.051
W:	0.000	P: 0.000
Z:	.071	
R:	-81.444	
Mode:	TCP XY	
Setup:	COMPLETE	
Reference:	COMPLETE	
Tool Log:	ENABLED	

5. Move the cursor to Mode and press F4, [CHOICE]. You will see a screen similar to the following.

```

1: TCP XY           5: QUICK XY
2: TCP XYZ         6: TCP ALIGN
3: TCP XYZWPR
4: TCP Z
TCPMate Menu

      X: .011      Y: -.051      Z: .071
      W: 0.000     P: 0.000     R: 0.000
Mode:           TCP XYZWPR
Setup:          COMPLETE
Reference:      COMPLETE
Tool Log:      ENABLED

```

6. Select the TCPMate mode: Move the cursor to one of the following TCPMate modes, and press ENTER:

- **To correct the TCP location for TOOL frame xy,** select TCP XY.
- **To correct the TCP location for TOOL frame xyz,** select TCP XYZ.
- **To correct the TCP location for TOOL frame location xyz and for orientation wp,** select TCP XYZWPR.
- **To correct the TCP location for TOOL frame z,** select TCP Z.
- **To correct the TCP location for TOOL frame xy with only two searches,** select QUICK XY. This mode is faster than the TCP xy mode, which requires four searches, but it does not provide redundancy to verify the correct TCP.
- **To correct TCP location for TOOL frame xyz but also check the TCP's orientation,** select ALIGN TCP. The TCP's wpr values are not updated. This mode is useful to check alignment of spot welding guns.

Note If you select TCP XYZWPR, correction for orientation around the z axis perpendicular to the plate is not performed.

It is important to note that the TOOL frame is relative to the touch plate, and that the TOOL frame z axis corresponds to the touch plate z axis. Because the touch plate is always aligned with its own x-y axis, regardless of the physical mounting of the plate in WORLD space, the WORLD y-z axis and the touch plate z axis both align with the WORLD x axis. It might be helpful to think in terms of correcting for the TOOL axes that correspond to PLATE xy, PLATE xyz, PLATE xyzwpr, and PLATE z.

For TOS Wrist Mode Only

The wrist of the robot must not be directly over the circle in the touch plate when performing z detection. If your particular tooling is configured so that the wrist is directly over the circle (such as when cylindrical tooling comes directly out of the center of the faceplate and is parallel to the faceplate z axis), then you must select TCP XY. In this case, if you need to detect z, you must reconfigure the tooling so that the wrist is offset by at least 15 degrees from the circle in the plate.

You will see a screen similar to the following.

```

TCPMate Menu

Current Tool:    [1]
      X: 214.359   Y: -111.097   Z: -14.319
      W: 89.338   P: 3.116     R: -81.444
TCP Offset[mm]: .088
      X: .011     Y: -.051     Z: .071
      W: 0.000   P: 0.000   R: 0.000
Mode:           TCP Z
Setup:          COMPLETE
Reference:      COMPLETE
Tool Log:       ENABLED
  
```

.1.2.3 TCPMate Setup and Calibration

Before you can use TCPMate, you must set up and calibrate the TCP. Refer to [Table –2](#) for listings and descriptions of the TCPMate TCP calibration items. Use [Procedure -3](#) to calibrate the TCPMate TCP.

Table –2. TCPMate TCP Calibration Items

ITEM	DESCRIPTION
Sensor Type Values: TOS Wrist, I/O, TOS & I/O, or TOS All Axes Default: TOS All Axes	This item indicates the kind of sensor used for contact detection. The three modes are: <ul style="list-style-type: none"> • TOS Wrist (Torque Observer Sensor on Wrist) - the contact torque will be monitored, and only the observed torque will be used to determine the contact position. When TOS Wrist is selected, no I/O connection is required for contact detection. • I/O - the specified I/O will be monitored and only the state of this input will be used to determine the contact position. Use the digital input port for contact detection with I/O. Refer to Digital Port item in this table for more information. • TOS & I/O - both torque and I/O will be monitored, and the first one to be detected will be used to determine the contact position. • TOS All Axes (Torque Observer Sensor, All Axes) - The contact torque will be monitored on all robot axes, and only the observed torque will be used to determine the contact position. This is the default value. When TOS All Axes is selected, no I/O connection is required for contact detection.

Table -2. TCPMate TCP Calibration Items (Cont'd)

ITEM	DESCRIPTION
<p>Digital Port</p> <p>Values: RDI, DI, WDI, or WSI</p> <p>Default: RDI</p>	<p>This item indicates the desired input port type. The port types are:</p> <ul style="list-style-type: none"> • RDI - Robot Digital Input • DI - normal Digital Input • WDI - Weld Digital Input • WSI - Wire Stick circuit Input The port type that you specify is followed by the desired port number that will go from OFF to ON when contact is detected. The WDI and WSI ports are typically used for arc welding. You will set this item up if you selected either I/O, or I/O & TOS, as the Sensor Type. You do not need to set this up if you selected TOS as the Sensor Type.
<p>Circuit Enable</p> <p>Values: RDO, DOUT, WDO, or WSO</p> <p>Default: RDO</p>	<p>This item indicates the circuit enable digital output port type. The port types are:</p> <ul style="list-style-type: none"> • RDO - Robot Digital Output • DOUT - normal Digital Output • WDO - Weld Digital Output • WSO - Wire Stick circuit Output The port type that you specify is followed by the desired port number that will be used to enable the I/O detection circuit. The WDO and WSO ports are typically used for arc welding. You will set this item up if you selected either I/O, or I/O & TOS, as the Sensor Type. Set the port number to 0 (zero) if you are not using this feature.
<p>DEVICE SETUP</p>	
<p>Two Circles</p> <p>Values: TRUE or FALSE</p> <p>Default: FALSE</p>	<p>This item enables the Two Circles method. If set to FALSE, Two Circles method is disabled.</p> <p>Note SpotTool+ uses only the Single Circle method. Two Circles should be set to FALSE for SpotTool+.</p> <p>For ArcTool, this item indicates whether you have one or two circles on the touch plate. Valid values are:</p> <ul style="list-style-type: none"> • TRUE - select this if you are using an arc torch and a wire to detect the TCP position. If TRUE, TCPMate will always execute a motion to detect the plate position (even if only XY is selected for an offset), so the tooling must be larger than the small circle. • FALSE - select this when the touch plate has only a single circle

Table -2. TCPMate TCP Calibration Items (Cont'd)

ITEM	DESCRIPTION
<p>Circle Axis</p> <p>Values: WORLD X, Y, Z, -X, -Y, -Z</p> <p>Default: WORLD Z</p>	<p>This item indicates the axis perpendicular to the touch plate. A normal installation is WORLD z, where -z motion from the home reference position will cause the tool to go through the touch plate circle.</p>
<p>Points on Circle</p> <p>Range: 4, 6, Or 8</p> <p>Default: 4</p>	<p>This item determines how many contact points are made between the tooling and the circular opening in the touch plate. Normally, four points are used to fit the change of TCP and to provide sufficient redundancy and error detection. However, if cycle time is not a constraint, and if you want best accuracy and reliability, choose six or eight points.</p>
<p>TCP Z Offset</p> <p>Units: mm</p> <p>Range: -500.000 - 500.000</p> <p>Default: 0.000</p>	<p>This item is the distance from the point on the tooling that contacts the plate to the actual TCP. This is used when XYZWPR mode is chosen to adjust the XYZ correction at the TCP properly when there is an orientation change. For spot welding and other applications where the contact point is the actual TCP location, this value should be zero. For Waterjet, arc welding, and other applications where the contact point is above the TCP at the time of contact, this value should be set to the distance from the contact point to the actual TCP.</p>
<p>Tool Log Enable</p> <p>Values: TRUE or FALSE</p> <p>Default: TRUE</p>	<p>This item enables logging of previous TCP values whenever a TCP update occurs. If set to FALSE, logging is disabled.</p>
<p>Tool Log Size</p> <p>Range: 5 - 100</p> <p>Default: 10</p>	<p>This item is the maximum number of TCP values that are stored in the tool log. When the maximum number is reached, the oldest stored TCP value is removed and the new one takes its place. This data is stored in permanent memory, so larger table sizes take up more space. Each entry takes approximately 40 bytes of permanent memory.</p>
<p>Auto Update</p> <p>Values: TRUE or FALSE</p> <p>Default: TRUE</p>	<p>This item indicates whether the automatic operation will cause the UTOOL to be changed. Automatic operation is used to detect when a collision has caused the tool to bend. It allows the Error Tolerance and Report On fields to determine whether production is stopped so that the problem can be fixed. If Auto Update is used, the UTOOL is always updated (unless Error Tolerance is exceeded).</p>

Table -2. TCPMate TCP Calibration Items (Cont'd)

ITEM	DESCRIPTION
Entering, Exiting prog Values: valid program name(s)	These items indicate the programs that will be used before and after TCPMate motion is executed during automatic operation. These items provide entrance and exit paths, ensure a safe path to the TCPMate Approach position, and ensure the proper state of the tooling or gripper before TCPMate operation. These programs must be set for automatic operation, and they can be the same program.
Calling Prog Values: valid program name	This item indicates the teach pendant program has the CALL TCPMATE instruction. TCPMATE will automatically increase the calling program's stack size if the program name is selected.
MOTION SETUP	
Failure Retry Values: 1 - 9 Default: 2	This item is the number of retries that will automatically perform if a failure occurs during TCPMate operation. For example, if excessive vibration causes a failure, TCPMate will continue to retry until either this number is exceeded, or until successful TCP correction is achieved.
Motion Speed Units: mm/sec Range: 25 - 500 Default: 50	This item is the positioning speed for TCPMate. This value can be increased if cycle time is of great concern, or it can be decreased if vibration or other disturbances cause problems with the TOS.
Fine Speed Units: mm/sec Range: 0.5 - 20 Default: 10	This item is the search speed for TCPMate. This speed is used for both TOS and I/O searches. The value can be increased if cycle time is of great concern (although increasing the speed will also affect the resultant accuracy and repeatability) , or reduced if vibration or other disturbances cause TCPMate to retry excessively.

Table -2. TCPMate TCP Calibration Items (Cont'd)

ITEM	DESCRIPTION
<p>Search Distance</p> <p>Units: mm</p> <p>Range: 50 - 500</p> <p>Default: 50</p>	<p>This item is the maximum distance that TCPMate will move while looking for the signal.</p>
<p>Learn Distance</p> <p>Units: mm</p> <p>Range: 0.5 – 10.0</p> <p>Default: 5.0</p>	<p>For“TOS All Axes” method only, there is a short motion before each search to identify dynamic torque values. Learn Distance is the distance in millimeters of this short motion, the direction is determined at the time of the motion.</p>
<p>Learn Speed</p> <p>Units: mm/sec</p> <p>Range: 1.0 – 75.0</p> <p>Default: 10.0</p>	<p>For“TOS All Axes” method only, this is the speed of the learning motion before each search.</p>
<p>Fit Tolerance</p> <p>Units: mm</p> <p>Range: 0.1 - 5.0</p> <p>Default: 0.25</p>	<p>This item is the circular fit tolerance for a successful TCP correction. If some error causes this tolerance to be exceeded, then the TCPMate operation will fail and will be done again up to the specified number of failure retries. This tolerance is the average point deviation from a perfect circle.</p>
<p>Radius Tolerance</p> <p>Units: mm</p> <p>Range: 0.1 - 5.0</p> <p>Default: 0.5</p>	<p>This item is a fit tolerance to the radius that was determined during TCPMate mastering. This item works similarly to Fit Tolerance. If you are using the same TOOL number for different diameters of tools (or significantly different weld wire diameters), you might want to increase this value.</p>

Table -2. TCPMate TCP Calibration Items (Cont'd)


ITEM	DESCRIPTION
Align Threshold Units: mm Range: 0.0 -5.0 Default: 2.0	This item is the maximum alignment change allowed in millimeters from the previous TCP to the new TCP. If the alignment change is greater than the slign threshold, then a prompt box will be displayed.
Error Tolerance Units: mm Range: 0.1 - 100 Default: 5.0	This item is the distance in millimeters from the previous TCP to the new TCP for a single correction. If a single correction is larger than Error Tolerance, then a prompt box will be displayed.
Report On Default: DOUT	This item is the digital output port that will be set to ON if the Error Tolerance value is exceeded during automatic operation. This allows the PLC to refrain from executing program motion until the problem is resolved.
DISTURBANCE TORQUE SETUP	
Sensitivity Tolerance Units: max. motor torque/sec. Range: 1 - 999 Default: 10	This item is the primary sensitivity tolerance of TOS. It is the rate of change of the disturbance torque. The actual units vary with motor model, gear ratio, and amplifier current, and are affected by tooling length. However, the effect for TCPMate is largely independent of these factors and a value of 10 is considered normal operation for typical installations. You should not use a value of less than 10 or misdetections might occur. This value can be increased if false detections occur frequently.
	<div style="border: 1px solid black; padding: 5px;">  <p>Warning</p> <p>Be careful when you change this number. Changing the sensitivity tolerance might cause a false detection. This could injure personnel or damage equipment.</p> </div>

Table -2. TCPMate TCP Calibration Items (Cont'd)

ITEM	DESCRIPTION
Torque Threshold Units: max. motor torque Range: 1 to 999 Default: 100	This item is the level of disturbance torque that is used as another means of TCPMate detection. The default value of 100 is considered normal operation for typical installations. If increasing the value of Sensitivity Tolerance does not make false detections disappear, then this value should be increased until the false detections disappear; then, reset the value of Sensitivity Tolerance .
Accumulate Check Values: TRUE or FALSE Default: FALSE	This item enables the Accumulate Check function, which causes TCPMate to post an error if the total accumulated TCP offset since mastering exceeds a specified amount.
Accumulate Threshold Units: mm Range: 1.0 – 100.0 Default: 5.0	This item sets the threshold for the total accumulated TCP offset in millimeters since TCPMate mastering before an error is posted. This function works if Accumulate Check is set to TRUE.

Procedure -3 Calibrating the TCPMate TCP

Conditions

- TCPMate mode is set on the TCPMate main menu. Refer to [Procedure -2](#).

Steps

1. Press **MENUS**.
2. Select **UTILITIES**.
3. Press **F1**, [**TYPE**].
4. Select **TCPMate**. You will see a screen similar to the following.

```

TCPMate Setup Menu
 1 Sensor Type:          TOS All Axes
 2 Digital Port:         RDI [ 1]
 3 Circuit Enable:       DOUT[ 1]
    Device Setup
 4 Two Circles:          FALSE
 5 Circle Axis:          World Z
 6 Points on Circle:     4
 7 TCP Z Offset:         30.000 mm
 8 Tool Log Enable:      TRUE
 9 Tool Log Size:        10
10 Auto Update:          TRUE
11 Entering prog:        *****
12 Exiting prog:         *****
13 Calling prog:         *****
    Motion Setup
14 Failure Retry:        2
15 Motion Speed:         50.000 mm/sec
16 Fine Speed:           10.000 mm/sec
17 Search Distance:      50.000 [mm]
18 learn speed:          20.000 mm/sec
19 learn distance:       2.500 [mm]
20 Fit Tolerance:        .250 [mm]
21 Radius Tolerance:     .500 [mm] ]
22 Align Threshold:      2.000 [mm]
23 Error Tolerance:      5.000 [mm]
24 Report On:            DOUT[ 0]
    Disturbance Torque Setup
25 Sensitivity Slope     10.000
26 Torque Threshold:     100]
27 Accumulate Error Check : TRUE
28 Accumulate Threshold: .010 [mm]

```

5. Move the cursor to Sensor Type and press F4, [CHOICE]. You will see a screen similar to the following.

```

1 TOS Wrist
2 I/O
3 I/O & TOS
4 TOS All Axes
  TCPMate Setup Menu
  1 Sensor Type:          TOS All Axes
  2 Digital Port:         RDI [ 1]
  3 Circuit Enable:       DOUT[ 1]
    Device Setup
  4 Two Circles:          TRUE
  5 Circle Axis:          World Z

```

Note The default sensor type is TOS All Axes (Torque Observer Sensor, All Axes). Normally, you will use TOS only. However, TCPMate can work with other sensors, such as electrical continuity.

6. Select the appropriate sensor type and press ENTER. Refer to [Table –2](#) for information on valid Sensor Type values.
7. **If you are using ArcTool**, perform the following steps. Otherwise, skip to [Step 8](#).
 - a. Move the cursor to Two Circles and press either F4, TRUE, or F5, FALSE. Refer to [Table –2](#) for information on valid Two Circles values.
 - b. If you selected I/O or I/O & TOS for the sensor type in [Step 6](#), move the cursor to Digital Port and press F4, [CHOICE]. You will see a screen similar to the following.

```
1 RDI
2 DI
3 WDI
4 WSI
TCPMate Setup Menu

1 Sensor Type:          TOS All Axes
2 Digital Port:        RDI [ 1]
3 Circuit Enable:      DOUT[ 1]
  Device Setup
4 Two Circles:         FALSE
5 Circle Axis:         World Z
```

- c. Select the desired input port type and press ENTER. Then, type the desired port number that will change from OFF to ON when contact is detected.

8. **If you selected I/O for the sensor type in [Step 6](#) and an output signal is required to enable the electrical continuity circuit**, move the cursor to Circuit Enable and press F4, [CHOICE].

```
1 RDO
2 DOUT
3 WDO
4 WSO
TCPMate Setup Menu

1 Sensor Type:          TOS All Axes
2 Digital Port:        RDI [ 1]
3 Circuit Enable:      DOUT[ 1]
  Device Setup
4 Two Circles:         TRUE
5 Circle Axis:         World Z
```

9. Select the desired output type and press ENTER. Then, type the desired port number that will change from OFF to ON to enable the continuity circuit before TCPMate motion starts.

After all search motion is complete, TCPMate will turn the port from ON to OFF.

-
- 10. If you have installed the TCPMate touch plate in anything other than the WORLD XY plane with -z motion going from the reference position through the circle, move the cursor to Circle Axis and press F4, [CHOICE]. You will see a screen similar to the following.**

```
1 World X           5 World -Y
2 World Y           6 World -Z
3 World Z           7
4 World -X          8
TCPMate Setup Menu
5 Circle Axis:      World Z
6 Points on Circle: 4
7 TCP Z Offset:    30.000 mm
8 Tool Log Enable: TRUE
9 Tool Log Size:   10
```

- 11.** Select the appropriate axis and press ENTER.
- 12.** Move the cursor to other items that you would like to set up, and specify the appropriate value according to [Table -2](#).

TCPMate setup is now complete.

.1.2.4 TCPMate Reference Positions

Before you can use TCPMate, you must record the reference positions. Use [Procedure -4](#) to record the reference positions.

Procedure -4 Recording TCPMate Reference Positions

Conditions

- If your sensor type is I/O, make sure the I/O port is working correctly.
- You have calibrated TCPMate. Refer to [Procedure -3](#).

Note Mastering the TCP referencing positions should be done only:

- The first time after setting the TCP
- After the robot is remastered
- If the Touch Plate is moved.

All other times, TCPMate should be run to recover the TCP when it is out of alignment.

Steps

1. Press MENUS.
2. Select UTILITIES.

3. Press F1, [TYPE].

4. Select TCPMate. You will see a screen similar to the following.

TCPMate Menu

```
Current Tool:  [1]
  X: 214.359   Y: -111.097   Z: -14.319
  W: 89.338    P: 3.116     R: -81.444
TCP Offset[mm]: .088
  X: .011      Y: -.051       Z: .071
  W: 0.000    P: 0.000     R: 0.000
Mode:         TCP XY
Setup:        COMPLETE
Reference:    COMPLETE
Tool Log:     ENABLED
```

5. Move the cursor to Reference and press ENTER.

Note If you are using an arc welding application and are using the wire to contact the contact plate, you need to set the Two Circles value to TRUE **before** you record a reference position.

- If you selected TCP XY or QUICK XY mode in [Procedure -3](#), you will see a screen similar to the following.

TCPMate Reference

```
1 Approach Position:  RECORD
2 Center Position 1:  RECORD
3 Plate Position:     RECORD
```

- If you selected TCP XYZ mode in [Procedure -3](#) and you have set **Two Circles to FALSE**, you will see a screen similar to the following.

TCPMate Reference

```
1 Approach Position:  RECORD
2 Center Position 1:  RECORD
3 Center Position 2:  RECORD
4 Plate Position:     RECORD
```

- If you selected TCP XYZWPR or TCP ALIGN mode in [Procedure -3](#) and you have set **Two Circles to FALSE**, you will see a screen similar to the following.

TCPMate Reference

```
1 Approach Position:  RECORD
2 Center Position 1:  RECORD
3 Center Position 2:  RECORD
4 Plate Position:     RECORD
```

-
- If you selected TCP XYZWPR mode (not available for SpotTool+) in [Procedure -3](#) and you have set **Two Circles to TRUE**, you will see a screen similar to the following.

```
TCPMate Reference
1 Approach Position:   RECORD
2 Small Circle Center: RECORD
3 Large Circle Center: RECORD
4 Plate Position:     RECORD
```

- If you selected TCP Z mode in [Procedure -3](#) , you will see a screen similar to the following.

```
TCPMate Reference
1 Plate Position:     RECORD
```

Note For TCP Z only, the approach position can be over any flat rigid surface. The robot will move in the direction specified by the Circle Axis value in the setup menu until it contacts the plate (refer to [Procedure -3](#) for information on TCPMate calibration). The circle is not used in this mode, and the robot must not be above a circle that will allow the tool to pass through it.

Note If you are running ArcTool and are **not** using the weld wire to detect contact, perform [Step 6](#). Otherwise, go to [Step 7](#).

- 6. To teach the reference for the "One Circle" method**, perform the following steps.
- a. Move the cursor to Approach Position and teach the position so that the tool is above the circle on the touch plate and is high enough so that the robot can move freely in the plane of the touch plate.
 - b. Press F3, RECORD, to record the position.
 - c. Move the cursor to Center Position 1 and teach the position so that the tool tip is within the circle on the touch plate and is roughly centered inside the circle. The TCP can extend a few millimeters through the hole so that if a collision causes the tool to bend, it will still contact the circle during TCPMate motion.
 - d. Press F3, RECORD, to record the position.
 - e. **If you selected XYZWPR or TCP ALIGN mode (refer to [Step 6 in Procedure -3](#)),** move the cursor to Center Position 2 and teach the position so that the tool shaft extends several inches (or as much as possible) through the circle while the cylindrical shaft is still centered in the circle.
 - f. Press F3, RECORD, to record the position.
 - g. **If you selected XYZ, XYZWPR, or TCP ALIGN mode**, move the cursor to the plate position and teach the position so that the tool is above the plate and is outside of the circle. Make sure the tool can make contact with the plate when the robot is doing -Z search.
 - h. Press F3, RECORD, to record the position.

-
- i. When you are finished recording positions, press F2, MASTER. You will see a prompt box similar to the following.

```
TCPMate Reference  
  
WARNING!  Robot Will Move If Continued  
  Press F2 to Continue  
  Press F3 to Abort  
  
          CONT      ABORT
```

- j. Verify that all cell fences and other safety devices are set for automatic operation and press F2, CONT, to continue.

Note If the operation was unable to complete, verify that the positions were taught close to the center of the circles and that all tooling and fixturing is securely anchored.

- k. Press PREV to display the TCPMate main menu. You can now recover or verify a TCP in the event of a collision.

7. If you are running ArcTool and you are using the weld wire to detect contact, perform the following steps:

- a. Move the cursor to Approach Position and teach the position so that the tool is above the small circle and is roughly centered above the circle.
- b. Press F3, RECORD, to record the position.
- c. Move the cursor to Small Circle Center and teach the position so that the weld cup is several millimeters above the touch plate with the wire extending through the small circle.
- d. Press F3, RECORD, to record the position.
- e. If you selected XYZWPR mode (refer to [Step 6](#)), move the cursor to Large Circle Center and teach the position so that the tool shaft extends several inches (or as much as possible) through the circle while the cylindrical shaft is still centered in the circle.
- f. Press F3, RECORD, to record the position.

Note After teaching the Large Circle Center, it is important to jog the robot out of the circle before mastering. When you master, the robot will move directly from the current position to the Approach Position defined in [Step 7a](#).

- g. When you are finished recording positions, press F2, MASTER. You will see a prompt box similar to the following.

TCPMate Reference

WARNING! Robot Will Move If Continued
Press F2 to Continue
Press F3 to Abort

CONT ABORT

- h. Verify that all cell fences and other safety devices are set for automatic operation and press F2, CONT, to continue.

Note If the operation was unable to complete, verify that the positions were taught close to the center of the circles and that all tooling and fixturing is securely anchored.

- i. Press the PREV key to return to the TCPMate main menu. You can now recover or verify a TCP in the event of a collision.

.1.3 TCPMate Troubleshooting

To troubleshoot TCPMate, you can

- Display the TOOL Log screen
- Execute Motion to check the TCP

.1.3.1 Tool Log Screen

The Tool log is a buffer that holds TOOL frame information. The buffer can hold up to as many items as are specified in the setup menu. Refer to [Procedure -3](#). When the buffer is full, it will replace the oldest data in the buffer with the new logged data. If you want to view TOOL frame information, you can display the TOOL Log screen. Use [Procedure -5](#) to display the TOOL Log screen.

Procedure -5 Displaying the TOOL Log Screen

Steps

1. Press MENU.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select TCPMate. You will see a screen similar to the following.

TCPMate Menu

```
Current Tool: [1]
  X: 214.359   Y: -111.097   Z: -14.319
  W: 89.338   P: 3.116     R: -81.444
TCP Offset[mm]: .088
  X: .011     Y: -.051     Z: .071
  W: 0.000   P: 0.000   R: 0.000
Mode:          TCP XY
Setup:         COMPLETE
Reference:     COMPLETE
Tool Log:      ENABLED
```

5. Move the cursor to Tool Log and press F4, [DETAIL], or ENTER. You will see a screen similar to the following.

TCPMate Tool Log

Date	Time	X(W)	Y(P)	Z(R)
01-JAN-XX	16:57	214.1	-110.6	-13.4
Tool Num [1]		93.8	-21.0	-81.9
01-JAN-XX	16:57	214.1	-110.6	-13.5
Tool Num [1]		89.3	3.1	-81.4
01-JAN-XX	16:57	214.1	-110.6	-13.5
Tool Num [1]		89.3	3.1	-81.4
01-JAN-XX	16:58	214.1	-110.6	-13.5
Tool Num [1]		89.3	3.1	-81.4
01-JAN-XX	16:58	214.1	-110.5	-13.4

Each entry spans two lines. The first line lists the date and time of the saved entry as well as x, y, and z information. The second line contains the tool number and w, p, and r information.

If entries are not used or if they have been cleared, you will see the following lines:

NO XYZ DATA

NO WPR DATA

6. Select one of the following:

- If you want to save the entire log to the file "TCP_LOG.DT," press F2, SAVE.
- If you want to restore a previous UTOOL value, move the cursor to the line that has the values you want to restore and press F3, APPLY. You will see a prompt box similar to the following.

```
TCPMate Tool Log
```

```
Update the UTOOL?
```

```
Please make sure to verify  
robot paths before going  
into production.
```

```
YES NO
```

Move the cursor to YES and press ENTER.

- **If you want to clear the log and remove all data entries**, press F4, CLEAR. You will see a screen similar to the following.

```
TCPMate Tool Log
```

```
Clear the whole tool  
log buffer?
```

```
YES NO
```

Move the cursor to YES and press ENTER.

.1.3.2 TCPMate Execute Screen

If you want to execute motion, you can do so from the TCPMate Execute screen. You might want to execute motion to verify that the TOOL is valid, or if you suspect that the physical tooling has been changed due to collision, repair, or other reasons. Use [Procedure -6](#) to execute motion from the TCPMate Execute screen.

Procedure -6 Executing Motion from the TCPMate Execute Screen

Conditions

- TCPMate mastering is complete. Refer to the “Recording Reference Positions” section of [Procedure -2](#).
- The cell is prepared for automatic operation.
- The robot has a clear travel path to the Approach position.

Steps

1. Press MENU.
2. Select UTILITIES.
3. Press F1, [TYPE].

4. Select TCPMate. You will see a screen similar to the following.

```
TCPMate Menu

Current Tool:   [1]
      X: 214.359   Y: -111.097   Z: -14.319
      W: 89.338    P: 3.116     R: -81.444
TCP Offset[mm]: .088
      X: .011      Y: -.051      Z: .071
      W: 0.000     P: 0.000     R: 0.000
Mode:          TCP XY
Setup:         COMPLETE
Reference:     COMPLETE
Tool Log:      COMPLETE
```

5. Press F2, EXEC. You will see a screen similar to the following.

```
Warning Message

      START TCPMATE MOTION?

WARNING!  Robot Will Move If Continued

      Press F2 to Continue
      Press F3 to Abort
```

- **If you want to cancel this operation without executing motion,** press F3, ABORT.
- **If you want to execute TCPMate motion,** press F2, CONT. When the operation completes, you will see a screen similar to the following with the new corrections to the Tool frame and a message in the status line.

```
Motion Done.  Press Update to update tool.
TCPMate Menu

Current Tool:   [1]
      X: 214.359   Y: -111.097   Z: -14.319
      W: 89.338    P: 3.116     R: -81.444
TCP Offset[mm]: .088
      X: .011      Y: -.051      Z: .071
      W: 0.000     P: 0.000     R: 0.000
Mode:          TCP XY
Setup:         COMPLETE
Reference:     COMPLETE
Tool Log:      COMPLETE
```

- **If you want to apply the correction to the current TOOL frame,** press F3, UPDATE.

Note If the error tolerance is exceeded during automatic operation, you will see a prompt box similar to the following.

```
TCPMate Menu

The new TCP is differed from
the current TCP by 2.144 [mm].
Please verify the TCP before
using it on existing programs.

OK
```

This prompt box is a warning that the UTOOL is about to be updated. If you specified an output port in the Report On field on the SETUP menu, that port will be set to ON and will stay ON until you press ENTER to close the above prompt box.

.1.4 TCPMate Adjustment

You can perform a TCPMate adjustment using either of the following methods:

- **Manually**, by pressing F2, EXEC, on the TCPMate main menu. When TCPMate is finished calculating the tool offset, press F3, UPDATE, to accept the change.
- **Automatically**, by incorporating the TCPMATE adjustment instruction into a teach pendant program. The setup menu must have "Auto Update" set to TRUE for the TCPMate adjustment to become effective automatically.

Note If you have a large TCP correction (greater than 3mm XYZ or 0.5 degrees orientation change), for best accuracy of TCP recovery you should run TCPMate a second time after executing TCPMate and updating the TCP.

.1.4.1 Manual Adjustment

After you have set up TCPMate, you can use it at any time to compensate for a misaligned TCP due to tool body damage. Then, you can monitor TCPMate adjustments to review the tool offsets made each time a TCPMate adjustment is performed.

Use [Procedure -7](#) to perform a manual TCPMate adjustment.

Procedure -7 Manual TCPMate Adjustment

Conditions

- You have performed all required TCPMate setup. Refer to [Procedure -2](#).

- **If you have selected I/O or I/O & TOS as the TCPMate sensor type**, all necessary circuitry should be enabled. Refer to [Procedure -3](#).
- Your program has been created using a TOOL frame that has been mastered with TCPMate. Refer to [Procedure -4](#).
- A worn or damaged tool is causing a problem in program execution or performance.
- The robot has a clear linear path to the touch plate. The robot is not required to be at the reference position to run the TCPMATE program. The position should be located in the notch of the touch plate.

Note If you did not perform all TCPMate setup before the tool became worn or damaged, then you cannot use TCPMate to compensate for a mislocated TCP.

Steps

1. Confirm that the welding tool is causing the problem.

Table -3. Confirming the Cause of the Problem

Use TCPMate if	Do Not Use TCPMate if
<ul style="list-style-type: none"> • The tool body is bent. • To verify the TCP. • To verify the robot calibration. 	<ul style="list-style-type: none"> • The robot does not return to zero reference marks. • Peripheral equipment is interfering with the tool. • Weld cable suspension devices are pulling back on the tool causing apparent mislocation. • Part nesting or clamping varies from part to part. • There is excessive cast in the welding wire, causing the wire to “flip” during operation (for ArcTool only).

2. Select the TCPMate main menu:
 - a. Press MENU.
 - b. Select UTILITIES.
 - c. Press F1, [TYPE].
 - d. Select TCPMate. You will see a screen similar to the following.

TCPMate Menu

```
Current Tool:  [1]
  X: 214.359   Y: -111.097   Z: -14.319
  W: 89.338   P: 3.116     R: -81.444
TCP Offset[mm]: .088
  X: .011     Y: -.051     Z: .071
  W: 0.000   P: 0.000   R: 0.000
Mode:         TCP XY
Setup:        COMPLETE
Reference:    COMPLETE
Tool Log:     ENABLED
```

3. To perform the manual adjustment, press F2, EXEC.
4. To accept the tool offset, press F3, UPDATE.

.1.4.2 Automatic Adjustment

If you want to adjust the TCP automatically, you can incorporate the TCPMATE program into your teach pendant program. The operation sequence is identical to a manual adjustment except that you do not have to select or start the TCPMATE program.

Use [Procedure -8](#) to perform an automatic TCPMate adjustment.

Procedure -8 Automatic TCPMate Adjustment

Conditions

- You have performed all required TCPMate setup. Refer to [Section 1.3](#).
- Your program has been created using a TOOL frame that has been mastered with TCPMate. Refer to [Procedure -4](#).
- A worn or damaged welding tool is causing a problem in program execution or weld performance.
- The robot has a clear linear path to the touch plate. The robot is not required to be at the TCP pointer for TCPMATE.

Note If you did not perform all TCPMate setup before the tool became worn or damaged, then you cannot use TCPMate to compensate for a misaligned TCP.

Steps

1. Insert the line "CALL TCPMATE" into your teach pendant program according to the example in [Automatic Adjustment Program Example](#) . Refer to for information on writing and modifying a program.

Automatic Adjustment Program Example

```
20: J P[10:HOME] 100% FINE      Original program
21: R[1] = R[1] + 1           Added instructions
22: IF R[1] = 10 JMP LBL1
23: JMP LBL2
24: LBL1
25: CALL TCPMATE
26: R[1] = 0
27: LBL2
[END]
```

2. Display the TCPMate setup menu:
 - a. Press MENUS.
 - b. Select UTILITIES.
 - c. Press F1, [TYPE].
 - d. Select TCPMate.
3. Move the cursor to select Calling Prog and select the teach pendant program name to increase the teach pendant program's stack size.

Note If you did not increase the stack size of the program, the first time the CALL TCPMATE statement is executed in your program you will see the following error message:

GUID-230 Not enough TPE stack

You can correct this condition in the following way:

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select TCPMate.

A message will be displayed indicating that the stack size has been increased automatically. You can now execute the program without the stack error.

Refer to the *FANUC Robotics SYSTEM R-J3iB Controller Error Code Manual* for information on other TCPMate error codes if they occur.

Refer to [Table 4](#) for information on how the TCPMATE program works according to the value of the Auto Update calibration item.

Table –4. Effects of Auto Update Value on the TCPMATE Program

If Auto Update is	And This Is True	Then This Will Occur
TRUE	--	The system variable \$MNUTOOL is updated every time the TCPMATE program is called.
	Logging is enabled	The previous TOOL is stored in the log.
	An error occurs	The program is paused and a prompt box is displayed.
	The change in TCP is greater than the Error Tolerance value as specified in the TCPMate SETUP menu	The TCP is not updated and a prompt box is displayed that explains the difference.
	The Error Tolerance value is exceeded and you have specified a DOUT port number in the Report On field in the TCPMate SETUP menu	The DOUT is set high.
FALSE	Each of the above items occur when Auto Update is set to FALSE with the following exception: the system variable \$MNUTOOL is NOT updated when the TCPMATE program is called. A notification message is displayed informing you that the current state of the tooling has changed (for example, a collision has occurred), but error correction will not be applied automatically.	

.1.5 TCPMate Error Recovery

If a TCPMate error occurs during TCP recovery, the robot goes back to the reference position and an error message is displayed. Review the error message, correct the problem, then restart TCPMate.

.1.5.1 Redoing a TCPMate Adjustment

To redo a TCPMate adjustment and restart TCPMate,

1. Review the error message for an indication on the problem.
2. Correct the problem.
3. Abort any program that is paused.
4. Press EXEC at the menu page (to display the TCPMate Menu screen, refer to [Procedure -2, Step 1](#) through [Step 4](#)). The robot will perform the TCPMate motion again.

Note The first time the CALL TCPMATE statement is executed in your program you will see the following error message: **GUID-230 Not enough TPE stack**

You can correct this condition in the following way:

1. Press MENUS.
2. Select UTILITIES.
3. Press F1, [TYPE].
4. Select TCPMate.

A message will be displayed indicating that the stack size has been increased automatically. You can now execute the program without the stack error.

Refer to the *FANUC Robotics SYSTEM R-J3iB Controller Error Code Manual* for information on other TCPMate error codes if they occur.

.1.5.2 TCPMate Fit or Convergence Errors

Refer to [Table -5](#) for information on troubleshooting TCPMate fit or convergence errors.

Table -5. TCPMate Fit and Convergence Errors

Try This	Description
Verify that the tool is rigid.	Worn or damaged tooling components can cause variation in tool repeatability and can cause fit errors.
Verify the payload setting.	If the reference position is at an extreme position, improper payload settings can cause increased robot vibration, which affects TCPMate repeatability.
Check for a loose touch plate. *	If the touch plate becomes loosened due to improper installation, TCPMate repeatability will be affected.

Table -5. TCPMate Fit and Convergence Errors (Cont'd)

Try This	Description
Reduce motion speed. *	Under certain conditions, the default motion speed can cause too much vibration. For best performance, TCPMate uses increased acceleration values. Try lowering the motion speed from the default 100mm/sec to 25 or 50mm/sec.
Reduce search speed. *	Under certain conditions, the default motion speed can cause too much vibration. For best performance, TCPMate uses increased acceleration values. Try lowering the search speed from the default 10mm/sec to 5mm/sec.
Increase the number of points on the circle. *	You can increase the number of points on the circle from 4 to either 6 or 8 to improve TCPMate repeatability.
Increase the retry count. *	TCPMate has significant safeguards against getting incorrect TCP corrections. You can increase the retry count from the default of 2 to either 3 or 4 to reduce the occurrence of TCPMate fit errors.
Increase the radius tolerance. *	The default radius tolerance of .5mm may not be large enough to allow for the normal variation of some tooling. You can try increasing the radius tolerance from .5mm to 1.0mm or more as your application allows.
Increase the fit tolerance.*	Most TCPMate errors are detected by radius tolerance, but if increasing the radius tolerance does not resolve the problem, you can increase the fit tolerance from .25mm to .5mm or more as your application allows.
If you are using TOS, increase the sensitivity tolerance value. *	Normally, the default value of 10 is adequate; however, you can increase this value to either 15 or 20.
If you are using TOS, increase the torque threshold. *	Normally, the default value of 100 is adequate. However, if the sensitivity tolerance increases have no effect, you can increase this value to 150.

* Redo TCPMate mastering after making this change. Refer to [Section .1.2.4](#).

Purpose of this Manual

This manual describes **FANUC Robotics HandlingTool setup and operation:**

- Setup
- Program development and testing
- Production run
- Status display
- Error recovery

How to Use this Manual

Use this table to locate specific information in the manual.

If you want to	Refer to
Find information about a specific topic	Table of Contents or Index
Review the characteristics of the HandlingTool system	Chapter 1 , Overview
Turn on, off, and jog the robot	Chapter 2 , Turning On and Jogging the Robot
Set up general information	Chapter 3 , General Setup
Set up system configuration	Chapter 4 , System Configuration Setup
Plan, create, and modify an application program	Chapter 5 , Planning and Creating a Program
Look up detailed information about a specific program instruction	Chapter 6 , Program Elements
Test a program, pause and restart a program, run a program, run production, and make adjustments during program operation	Chapter 7 , Testing a Program and Running Production
View status information on teach pendant screens and using other indicators	Chapter 8 , Status Displays and Indicators
Copy, rename, delete, load, and transfer files	Chapter 9 , Program and File Manipulation
Use mirror shift, program base shift, and other advanced programming functions	Chapter 10 , Advanced Functions
Use the FANUC sensor interface	Chapter 11 , FANUC Sensor Interface
Use the Auto TCP function	Chapter 12 , Automatic Tool Center Point
Use error messages and recovery procedures to solve problems	Appendix A , Error Codes and Recovery
Use the optional CRT/KB	Appendix B , CRT/KB Setup and Operation
Use controller initialization utilities	Appendix C , BootROM Operations
Review program examples	Appendix D , Program Examples
Master the robot	Appendix E , Mastering
Review robot transportation information	Appendix F , Transportation and Installation

Conventions Used in this Manual

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



WARNING

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



CAUTION

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

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

Preface	vii
Safety	xxxvii

Chapter 1

OVERVIEW	1-1
1.1 ROBOT	1-3
1.1.1 Robot Models	1-4
1.1.2 End-of-Arm Tooling	1-13
1.1.3 Extended Axes	1-13
1.2 CONTROLLER	1-14
1.2.1 Teach Pendant	1-17
1.2.2 Standard Operator Panel (SOP)	1-20
1.2.3 User Operator Panel (UOP)	1-26
1.2.4 CRT/KB	1-26
1.2.5 Emergency Stop Devices	1-26
1.2.6 Robot Stop Variation (for European Controllers)	1-27
1.2.7 Robot Stop Variation (for Control Reliable (RS-1/RS-4) option only)	1-28
1.2.8 Communications	1-29
1.2.9 Input/Output (I/O)	1-30
1.2.10 Remote I/O Interfaces	1-30
1.2.11 Motion	1-31
1.2.12 Extended Axes	1-31
1.2.13 Controller Backplane	1-32
1.2.14 Memory	1-32
1.3 HANDLINGTOOL SOFTWARE	1-33
1.3.1 Set Up	1-33
1.3.2 Program	1-33
1.3.3 Test Program	1-35
1.3.4 Run Production	1-35
1.4 MENU MAPS	1-36
1.4.1 UTILITIES Menu Map	1-37
1.4.2 TEST CYCLE Menu Map	1-39
1.4.3 MANUAL FCTNS Menu Map	1-40
1.4.4 ALARM Menu Map	1-40
1.4.5 I/O Menu Map	1-41
1.4.6 SETUP Menu Map	1-43
1.4.7 FILE Menu Map	1-49
1.4.8 USER Menu Map	1-50
1.4.9 SELECT Menu Map	1-50
1.4.10 EDIT Menu Map	1-51
1.4.11 DATA Menu Map	1-52

1.4.12 STATUS Menu Map	1-53
1.4.13 POSITION Menu Map	1-54
1.4.14 SYSTEM Menu Map	1-54

Chapter 2

TURNING ON AND JOGGING THE ROBOT	2-1
2.1 TURNING ON AND TURNING OFF THE ROBOT	2-2
2.2 JOGGING THE ROBOT	2-5
2.2.1 Jogging when Activation from the Teach Pendant is Inhibited	2-5
2.2.2 Jog Speed	2-6
2.2.3 Coordinate Systems	2-7
2.2.4 Wrist Jogging	2-9
2.2.5 Remote TCP Jogging (option)	2-9
2.2.6 Motion Groups	2-10
2.2.7 Extended Axes and Sub-Groups	2-10
2.2.8 Jog Menu	2-15

Chapter 3

GENERAL SETUP	3-1
3.1 INPUTS AND OUTPUTS (I/O) SETUP	3-3
3.1.1 Analog I/O	3-4
3.1.2 Digital I/O	3-10
3.1.3 Group I/O	3-18
3.1.4 Robot I/O	3-23
3.2 I/O INTERCONNECT SETUP	3-28
3.3 USER OPERATOR PANEL (UOP) I/O SIGNALS	3-33
3.3.1 UOP Input Signals	3-35
3.3.2 UOP Output Signals	3-39
3.4 MODEL B I/O SETUP	3-43
3.4.1 Setting the DIP Switches	3-46
3.4.2 Setting Up the Basic Digital I/O Units	3-49
3.4.3 Setting Up User I/O	3-49
3.4.4 Digital I/O	3-50
3.4.5 Group I/O	3-58
3.5 PLC I/O SETUP	3-63
3.6 I/O LINK SCREEN	3-70
3.6.1 I/O Link Device Screen	3-70
3.6.2 Model B I/O Detail Information	3-72
3.6.3 Setting Number of Ports	3-74

3.7 CONTROLLING I/O	3-75
3.7.1 Forcing Outputs	3-75
3.7.2 Simulating Inputs and Outputs	3-76
3.8 FRAMES SETUP	3-78
3.8.1 Setting Up Tool Frame	3-80
3.8.2 Setting Up User Frame	3-94
3.8.3 Setting Up a Remote TCP Frame	3-110
3.8.4 Setting Up Jog Frame	3-120
3.8.5 Saving Frame Data	3-129
3.9 PRODUCTION OPERATION SETUP	3-131
3.9.1 Robot Service Request (RSR)	3-131
3.9.2 Program Number Select (PNS)	3-134
3.10 MACRO COMMANDS	3-136
3.10.1 Setting Up Macro Commands	3-136
3.10.2 Executing Macro Commands	3-141
3.11 AXIS LIMITS SETUP	3-145
3.12 BRAKE TIMERS SETUP	3-148
3.13 BRAKE ON HOLD SETUP	3-151
3.14 CURRENT LANGUAGE SETUP	3-152
3.15 IGNORE OFFSET SETUP	3-153
3.16 IGNORE TOOL OFFSET SETUP	3-154
3.17 USER ALARM SETUP	3-155
3.18 OVERRIDE SELECT SETUP	3-158
3.19 PASSWORD SETUP	3-161
3.19.1 Install User Password Operations	3-163
3.19.2 Program and Setup User Password Operations	3-167
3.19.3 Password Log	3-170
3.19.4 Password Level Screen Permissions	3-173
3.20 ROBOT PAYLOAD SETTING	3-176
3.20.1 Payload Setting Process	3-176
3.20.2 Payload Setting Items	3-176
3.20.3 Payload Setup Procedure	3-178
3.20.4 Payload Teach Pendant Program Instruction	3-182
3.20.5 Inertia Equations	3-183

Chapter 4

SYSTEM CONFIGURATION SETUP	4-1
4.1 SYSTEM CONFIGURATION SETUP SCREEN	4-2
4.2 SYSTEM CONFIGURATION SETUP PROCEDURE	4-6

Chapter 5

PLANNING AND CREATING A PROGRAM	5-1
5.1 PLANNING A PROGRAM	5-2
5.1.1 Motion	5-2
5.1.2 Predefined Positions	5-4
5.2 WRITING AND MODIFYING A PROGRAM	5-9
5.2.1 Writing a New Program	5-10
5.2.2 Modifying a Program	5-17
5.3 MODIFYING A PROGRAM IN THE BACKGROUND (BACKGROUND EDITING)	5-30
5.3.1 Background Edit Process Flow	5-31
5.3.2 Troubleshooting Background Edit	5-35

Chapter 6

PROGRAM ELEMENTS	6-1
6.1 PROGRAM HEADER INFORMATION	6-5
6.1.1 Creation Date	6-5
6.1.2 Modification Date	6-5
6.1.3 Copy Source	6-5
6.1.4 Positions and Program Size	6-6
6.1.5 Program Name	6-6
6.1.6 Sub Type	6-7
6.1.7 Program Comment	6-7
6.1.8 Group Mask	6-8
6.1.9 Write Protection	6-8
6.1.10 Ignore Pause	6-9
6.2 LINE NUMBER AND PROGRAM END MARKER	6-10
6.3 MOTION INSTRUCTION	6-11
6.3.1 Motion Type	6-12
6.3.2 Positional Information	6-20
6.3.3 Frame Number of Positional Data	6-21
6.3.4 Speed	6-23
6.3.5 Termination Type	6-32
6.3.6 Motion Options	6-34
6.3.7 AccuPath	6-47
6.3.8 Correspondence between Teach Pendant Program Motion and KAREL Program Motion ..	6-57
6.4 INTELLITRAK	6-58
6.5 PALLETIZING INSTRUCTIONS	6-60
6.5.1 PALLETIZING-B Instruction	6-62
6.5.2 PALLETIZING-BX Instruction	6-68
6.5.3 PALLETIZING-E Instruction	6-75

6.5.4 PALLETIZING-EX Instruction	6-82
6.5.5 PALLETIZING-END Instruction	6-90
6.6 PALLET REGISTER INSTRUCTIONS	6-91
6.7 REGISTER INSTRUCTIONS	6-93
6.8 POSITION REGISTER INSTRUCTIONS	6-96
6.8.1 PR[x] Position Register Instructions	6-96
6.8.2 PR[i,j] Position Register Element Instructions	6-97
6.9 INPUT/OUTPUT INSTRUCTIONS	6-100
6.9.1 Digital Input and Output Instructions	6-100
6.9.2 Robot Digital Input and Output Instructions	6-101
6.9.3 Analog Input and Output Instructions	6-103
6.9.4 Group Input and Output Instructions	6-104
6.10 BRANCHING INSTRUCTIONS	6-105
6.10.1 Label Definition Instruction LBL[x]	6-105
6.10.2 Unconditional Branching Instructions	6-105
6.10.3 Conditional Branching Instruction	6-106
6.11 WAIT INSTRUCTIONS	6-109
6.12 MISCELLANEOUS INSTRUCTIONS	6-112
6.12.1 RSR Enable/Disable Instruction	6-112
6.12.2 User Alarm Instruction	6-112
6.12.3 Timer Instruction	6-113
6.12.4 OVERRIDE Instruction	6-113
6.12.5 Remark Instruction	6-113
6.12.6 Message Instruction	6-113
6.12.7 Parameter Name Instruction	6-114
6.12.8 Maximum Speed Instruction	6-118
6.13 SKIP INSTRUCTION SKIP CONDITION [I/O] = [VALUE]	6-119
6.14 OFFSET INSTRUCTIONS	6-121
6.15 TOOL OFFSET CONDITION INSTRUCTION	6-124
6.16 MULTIPLE CONTROL INSTRUCTIONS	6-125
6.17 MACRO COMMAND INSTRUCTION	6-126
6.18 PROGRAM CONTROL INSTRUCTIONS	6-127
6.18.1 PAUSE Instruction	6-127
6.18.2 ABORT Instruction	6-127
6.18.3 Error Program Instruction	6-128
6.18.4 Resume Program Instruction	6-128
6.19 SENSOR INSTRUCTIONS	6-129
6.20 MOTION GROUP INSTRUCTIONS	6-131
6.20.1 Independent Motion Group Instructions	6-131
6.20.2 Simultaneous Motion Group Instructions	6-131
6.20.3 Defining Motion Group Instructions	6-132
6.21 POSITION REGISTER LOOK-AHEAD INSTRUCTIONS	6-134
6.22 CONDITION MONITOR INSTRUCTIONS	6-135
6.23 PAYLOAD INSTRUCTION	6-137
6.24 COLLISION GUARD INSTRUCTIONS	6-139

Chapter 7

TESTING A PROGRAM AND RUNNING PRODUCTION	7-1
7.1 PROGRAM PAUSE AND RECOVERY	7-2
7.1.1 EMERGENCY STOP and Recovery	7-2
7.1.2 HOLD and Recovery	7-3
7.1.3 Setting Tolerance for Resuming a Program	7-4
7.2 TEST CYCLE	7-9
7.2.1 Test Cycle Setup	7-10
7.2.2 Single Step Testing	7-12
7.2.3 Continuous Testing	7-16
7.2.4 Monitoring Programs	7-20
7.3 RELEASE WAIT	7-21
7.4 PRODUCTION OPERATION	7-22
7.4.1 Standard Operator Panel Cycle Start Production	7-22
7.4.2 User Operator Panel Start	7-24
7.4.3 Robot Service Request (RSR) Production Start	7-25
7.4.4 Program Number Select (PNS) and UOP Production Start	7-27
7.5 ADJUSTING PROGRAM INFORMATION DURING PRODUCTION RUN	7-29
7.6 MAINTENANCE AND REPAIR	7-33

Chapter 8

STATUS DISPLAYS AND INDICATORS	8-1
8.1 STATUS INDICATORS	8-2
8.1.1 Teach Pendant Status Indicators	8-2
8.1.2 Standard Operator Panel Status Indicators	8-3
8.2 USER SCREEN STATUS	8-5
8.3 REGISTER STATUS	8-6
8.4 POSITION REGISTER STATUS	8-8
8.5 PALLET REGISTERS	8-11
8.6 SYSTEM VARIABLE STATUS	8-13
8.7 SAFETY SIGNAL STATUS	8-15
8.8 PROGRAM TIMER STATUS	8-18
8.9 SYSTEM TIMER	8-20
8.10 CLOCK	8-22
8.11 VERSION IDENTIFICATION STATUS	8-23
8.12 MEMORY STATUS	8-26
8.13 POSITION STATUS	8-28
8.14 TURN NUMBER DISPLAY	8-30
8.14.1 Usual Configuration	8-31
8.14.2 \$SCR_GRP[group] .turn_axis[i] System Variable	8-33

8.15 EXECUTION HISTORY	8-34
8.16 SOP I/O STATUS	8-36
8.17 DUTY DIAGNOSIS	8-38
8.17.1 Duty Value	8-38
8.17.2 Duty Status	8-39

Chapter 9

PROGRAM AND FILE MANIPULATION	9-1
9.1 STORAGE DEVICES	9-2
9.1.1 Setting Up a Port	9-3
9.1.2 Connecting a Disk Drive to the Controller	9-8
9.1.3 Using a Memory Card Interface	9-11
9.1.4 Setting the Default Device	9-17
9.1.5 Formatting Disks	9-19
9.2 MANIPULATING PROGRAMS	9-27
9.2.1 Selecting Programs on the SELECT Menu	9-28
9.2.2 Saving Programs to Disk	9-29
9.2.3 Loading Programs from Disk	9-31
9.2.4 Copying Programs Within the SELECT Menu	9-32
9.2.5 Deleting Programs from the SELECT Menu	9-34
9.2.6 Printing	9-35
9.3 MANIPULATING FILES	9-39
9.3.1 Generating a Directory of Files	9-41
9.3.2 Loading and Restoring Files from Disk To Controller Memory	9-44
9.3.3 Backing Up Program and System Files	9-53
9.3.4 Displaying Text (ASCII) Files	9-58
9.3.5 Copying Files to a Disk	9-59
9.3.6 Deleting Files from a Disk	9-61
9.3.7 Saving Files	9-63
9.3.8 Checking and Purging File Memory	9-65
9.4 CONTROLLER BACKUP AND RESTORE	9-66
9.4.1 Backing up a Controller	9-66
9.4.2 Restoring a Controller	9-72

Chapter 10

ADVANCED FUNCTIONS	10-1
10.1 MIRROR IMAGE UTILITY	10-4
10.2 PROGRAM SHIFT UTILITY	10-16
10.3 SPACE CHECK FUNCTION	10-26
10.4 EXECUTING MULTIPLE PROGRAMS (MULTI-TASKING)	10-30
10.4.1 Guidelines	10-30
10.4.2 Synchronizing the Execution of Multiple Programs	10-31
10.4.3 Affect of Multi-tasking on Dedicated I/O Signals	10-31
10.4.4 Standard Operator Panel (SOP) Cycle Start Execution	10-32
10.4.5 Program Number Select (PNS) Execution	10-33
10.4.6 RUN Program Instruction Execution	10-34
10.4.7 Single Step Program Execution	10-35
10.5 ANGLE ENTRY SHIFT FUNCTION	10-37
10.6 REFERENCE POSITION UTILITY	10-43
10.7 POSITION REGISTER LOOK-AHEAD EXECUTION FUNCTION	10-46
10.7.1 Program Instructions	10-47
10.7.2 Program Example	10-48
10.7.3 Execution	10-49
10.8 SHAPE GENERATION (OPTION)	10-50
10.8.1 Shape Setup	10-51
10.8.2 Shape Schedules	10-53
10.8.3 Programming	10-65
10.8.4 Teach and Production Modes	10-70
10.8.5 Shape Adjust Utility	10-71
10.8.6 Shape Frames	10-75
10.9 MOTION GROUP DO OUTPUT FUNCTION	10-76
10.9.1 Restrictions	10-76
10.9.2 Operations	10-76
10.9.3 Setup	10-77
10.9.4 Subprogram and Multi-tasking Execution	10-78
10.10 MOTION START DELAY DETECTION FUNCTION	10-79
10.10.1 Restrictions	10-79
10.10.2 Detection Information and System Variables	10-80
10.10.3 Motion Start Delay Detection Function Example	10-82
10.11 SOFT FLOAT FUNCTION (OPTION)	10-83
10.11.1 Soft Float Function Restrictions	10-84
10.11.2 Soft Float Schedules	10-86
10.11.3 Soft Float Program Instructions	10-90

10.12 CONTINUOUS TURN FUNCTION	10-92
10.12.1 Function	10-92
10.12.2 Setting	10-93
10.12.3 Motion Instruction	10-95
10.12.4 Operation	10-95
10.12.5 Example	10-96
10.12.6 Notes and Restrictions	10-97
10.12.7 Alarm Codes	10-98
10.13 CRT FUNCTION	10-99
10.13.1 Operation	10-102
10.13.2 Troubleshooting	10-105
10.14 SINGULARITY CHECK FUNCTION	10-106
10.15 ALL-POINT TEACHING FOR PALLETIZING	10-107
10.16 COORDINATES OFFSET FUNCTION	10-110
10.16.1 Tool Frame Offset Function	10-114
10.16.2 User Frame Offset Function	10-117
10.17 TIME BEFORE/AFTER MOTION OPTION INSTRUCTION	10-121
10.17.1 Program Execution	10-121
10.17.2 Execution Timing	10-122
10.17.3 Recording a TIME BEFORE/AFTER Instruction	10-123
10.17.4 TIME BEFORE Instruction Program Example	10-125
10.17.5 Programming Hints	10-126
10.18 CONDITION MONITOR FUNCTION	10-127
10.18.1 Monitors	10-128
10.18.2 Monitor State	10-129
10.18.3 Monitor Instructions	10-130
10.18.4 Condition Handler Program	10-130
10.18.5 Conditions	10-131
10.18.6 Condition Menu	10-132
10.18.7 Restrictions	10-134
10.19 COLLISION GUARD (OPTION)	10-141
10.19.1 Limitation	10-141
10.19.2 Falsely Detected Collisions	10-142
10.19.3 Collision Guard Adjust Macro Program	10-142
10.19.4 Setup	10-143
10.19.5 Programmed Motion	10-145
10.20 ERROR RECOVERY (OPTION)	10-146
10.20.1 Overview	10-146
10.20.2 Features	10-149
10.20.3 Limitations	10-150
10.20.4 I/O Interface	10-150
10.20.5 Setup	10-152

10.20.6	Programming	10-158
10.20.7	Testing	10-161
10.20.8	Error Recovery Manual Function	10-161
10.20.9	I/O Timing Sequence	10-164
10.21	AUTO NORMAL UTILITY (OPTION)	10-169
10.21.1	Overview	10-170
10.21.2	Setup	10-173
10.21.3	Execution	10-175
10.22	TCP SPEED PREDICTION (OPTION)	10-176
10.22.1	Overview	10-177
10.22.2	System Variables	10-179
10.23	PROGRAM TOOLBOX (OPTION)	10-184
10.23.1	Cross Car Mirror (option)	10-185
10.23.2	UTOOL Adjust (option)	10-189
10.23.3	Flip Knuckle (Option)	10-194
10.23.4	Limit Set (option)	10-197

Chapter 11

FANUC SENSOR INTERFACE	11-1
11.1 COMMUNICATION PROTOCOL	11-2
11.1.1 Flow Control Mechanism	11-2
11.1.2 Data Format	11-2
11.1.3 Hand Shaking	11-4
11.2 PROGRAMMING	11-6
11.3 SENSOR SETUP AND HARDWARE CONNECTIONS	11-7

Chapter 12

AUTOMATIC TOOL CENTER POINT	12-1
12.1 OVERVIEW	12-2
12.2 HARDWARE REQUIREMENTS AND INSTALLATION	12-3
12.2.1 Installing the String Sensor and TCP Attachment Device	12-3
12.2.2 Sensor I/O Signal Cables	12-5
12.2.3 Sensor Setup	12-7
12.2.4 Testing Data Ports	12-10
12.3 TCP ORIENTATION SETUP	12-11
12.4 AUTO SETUP	12-13
12.4.1 Tool Orientation	12-13
12.4.2 Automatically Generating Positions	12-14
12.4.3 Manually Defining Positions	12-14
12.4.4 Automatic and Manual Position Generation	12-14
12.5 AUTOMATICALLY CALCULATING A NEW UTOOL TCP	12-20

Appendix A

ERROR CODES AND RECOVERY	A-1
A.1 OVERVIEW	A-3
A.1.1 Facility Name and Code	A-6
A.1.2 Severity Descriptions	A-7
A.1.3 Error Message Text	A-9
A.2 GENERAL ERROR RECOVERY PROCEDURES	A-11
A.2.1 Overtravel Release	A-11
A.2.2 Hand Breakage Recovery	A-13
A.2.3 Pulse Coder Alarm Recovery	A-14
A.3 ERROR CODES	A-16

Appendix B

CRT/KB SETUP AND OPERATION	B-1
B.1 CRT/KB SETUP	B-2
B.2 CRT/KB MENUS	B-2
B.3 CRT/KB KEYS	B-2

Appendix C

BOOTROM OPERATIONS	C-1
C.1 STARTUP METHODS	C-2
C.1.1 INIT Start	C-2
C.1.2 Controlled Start (START CTRL)	C-3
C.1.3 Controlled 2 Start (START CTRL2)	C-6
C.1.4 Cold Start (START COLD)	C-7
C.1.5 Semi Hot Start	C-9
C.1.6 Re-Init Start (CMOSINIT)	C-10
C.2 BOOTROM UTILITIES	C-12
C.2.1 Extended Boot Monitor (EMON>) Utilities	C-15
C.2.2 Diagnostic Utilities	C-16
C.2.3 INSTALL Utilities	C-18
C.2.4 Flash ROM Utilities	C-20
C.2.5 Memory Card Utilities	C-21

Appendix D

PROGRAM EXAMPLES	D-1
D.1 /PROG PREG_ELE	D-2
D.2 /PROG PREG_VAL	D-2
D.3 REGISTER ANALOG INPUT	D-3
D.4 CONDITIONAL BRANCHING; USING LABELS	D-3
D.5 REGISTER INCREMENT	D-4
D.6 GROUP OUTPUT; WAIT INSTRUCTION PULSE INSTRUCTION	D-4
D.7 LABELS	D-5
D.8 LABEL; JUMP LABEL; MESSAGE	D-5
D.9 MACRO INSTRUCTION	D-6

Appendix E

MASTERING	E-1
E.1 RESETTING ALARMS AND PREPARING FOR MASTERING	E-3
E.2 MASTERING TO A FIXTURE (FIXTURE POSITION MASTER)	E-6
E.3 ZERO DEGREE MASTERING	E-8
E.4 SINGLE AXIS MASTERING	E-10
E.5 QUICK MASTERING	E-13

Appendix F

TRANSPORTATION AND INSTALLATION	F-1
F.1 TRANSPORTATION	F-1

Glossary **Glossary-1**

Index **Index-1**

List of Procedures

Procedure 2–1	Turning On the Robot	2–3
Procedure 2–2	Turning Off the Robot	2–4
Procedure 2–3	Jogging the Robot and Other Axes	2–11
Procedure 2–4	Using the Jog Menu	2–15
Procedure 3–1	Configuring Analog I/O – Rack, Slot, Channel	3–7
Procedure 3–2	Configuring Digital I/O – Rack, Slot, Start Point	3–13
Procedure 3–3	Configuring Digital I/O – Polarity and Complementary Pairs	3–16
Procedure 3–4	Configuring Group I/O – Rack, Slot, Start Point, Num Pts	3–21
Procedure 3–5	Configuring Robot I/O	3–24
Procedure 3–6	Using Interconnect I/O	3–29
Procedure 3–7	Configuring UOP I/O – Rack, Slot, Start Point	3–40
Procedure 3–8	Setting the DIP Switches on the Interface Unit	3–46
Procedure 3–9	Setting the DIP Switches on a Basic Digital I/O Unit	3–47
Procedure 3–10	Configuring Digital I/O – Rack, Slot, Start Point	3–52
Procedure 3–11	Configuring Digital I/O – Polarity and Complementary Pairs	3–55
Procedure 3–12	Configuring Group I/O – Rack, Slot, Start Point, Num Pts	3–60
Procedure 3–13	Configuring PLC I/O – Rack, Slot, Start Point	3–65
Procedure 3–14	Configuring PLC I/O – Polarity and Complementary Pairs	3–67
Procedure 3–15	Accessing the Model B I/O Detail Screen	3–72
Procedure 3–16	Forcing Outputs	3–75
Procedure 3–17	Simulating and Unsimulating Inputs and Outputs	3–76
Procedure 3–18	Setting Up Tool Frame Using the Three Point Method	3–82
Procedure 3–19	Setting Up Tool Frame Using the Six Point Method	3–85
Procedure 3–20	Setting Up Tool Frame Using the Direct Entry Method	3–90
Procedure 3–21	Selecting a Tool Frame	3–93
Procedure 3–22	Setting Up the User Frame Using the Three Point Method	3–95
Procedure 3–23	Setting Up User Frame Using the Four Point Method	3–100
Procedure 3–24	Setting Up User Frame Using the Direct Entry Method	3–105
Procedure 3–25	Selecting a User Frame	3–108
Procedure 3–26	Setting Up a Remote TCP Frame Using the Three Point Method	3–111
Procedure 3–27	Setting Up a Remote TCP Frame Using the Direct Entry Method	3–115
Procedure 3–28	Selecting an RTCP Frame	3–119
Procedure 3–29	Setting Up the Jog Frame Using the Three Point Method	3–121
Procedure 3–30	Setting Up the Jog Frame Using the Direct Entry Method	3–125
Procedure 3–31	Selecting a Jog Frame	3–128
Procedure 3–32	Saving Frame Data to a File	3–129
Procedure 3–33	RSR Setup	3–133
Procedure 3–34	PNS Setup	3–135
Procedure 3–35	Setting Up a Macro Command	3–139
Procedure 3–36	Executing a Macro Command from a Teach Pendant User Key	3–141
Procedure 3–37	Using the Teach Pendant Keys TOOL1 and TOOL2	3–143
Procedure 3–38	Executing a Macro Command from the MANUAL FCTNS Menu	3–144
Procedure 3–39	Setting Up Axis Limits	3–146
Procedure 3–40	Setting Brake Timers	3–148
Procedure 3–41	Setting Brake On Hold	3–151

Procedure 3-42	Setting Current Language	3-152
Procedure 3-43	Setting Ignore Offset	3-153
Procedure 3-44	Setting Ignore Tool Offset	3-154
Procedure 3-45	Setting User Alarm	3-156
Procedure 3-46	Setting User Alarm Severity	3-157
Procedure 3-47	Setting Up Override Select	3-160
Procedure 3-48	Assigning Usernames and Default Passwords for each Password Level	3-163
Procedure 3-49	Logging In	3-167
Procedure 3-50	Logging Out	3-168
Procedure 3-51	Changing Your Password	3-169
Procedure 3-52	Enabling the Password Log	3-171
Procedure 3-53	Displaying the Password Log	3-172
Procedure 3-54	Setting Robot Payload	3-178
Procedure 4-1	System Setting	4-6
Procedure 5-1	Creating and Writing a New Program	5-12
Procedure 5-2	Modifying a Program	5-22
Procedure 5-3	Modifying a Program in the Background	5-33
Procedure 6-1	Changing Motion Speed (from Speed Value to Register)	6-26
Procedure 6-2	Changing Motion Speed (from Register to Speed Value)	6-27
Procedure 6-3	Replacing Speed Values (using Motion Modify in [EDCMD] REPLACE)	6-28
Procedure 6-4	Using the Palletizing Editor To Enter the Palletizing-B Instruction	6-64
Procedure 6-5	Using the Palletizing Editor To Enter the Palletizing-BX Instruction	6-70
Procedure 6-6	Using the Palletizing Editor To Enter the Palletizing-E Instruction	6-77
Procedure 6-7	Using the Palletizing Editor To Enter the Palletizing-EX Instruction	6-85
Procedure 6-8	Defining a Parameter Name Instruction	6-115
Procedure 6-9	Defining Motion Group Instructions	6-132
Procedure 7-1	EMERGENCY STOP	7-2
Procedure 7-2	Recovery from EMERGENCY STOP	7-3
Procedure 7-3	HOLD and Recovery	7-3
Procedure 7-4	Setting Up Tolerance for Resuming a Program	7-6
Procedure 7-5	Resuming a Program that Exceeds the Stop Tolerance	7-7
Procedure 7-6	Setting Up Test Cycle Conditions	7-11
Procedure 7-7	Single Step Testing	7-14
Procedure 7-8	Continuous Testing Using the Teach Pendant	7-16
Procedure 7-9	Continuous Testing Using the Operator Panel CYCLE START Button	7-18
Procedure 7-10	Monitoring a Running Program	7-20
Procedure 7-11	Using Release Wait	7-21
Procedure 7-12	Running Production Using Standard Operator Panel (SOP) Cycle Start	7-23
Procedure 7-13	Running Production Using User Operator Panel (UOP) Start	7-24
Procedure 7-14	Running Production Using Robot Service Requests (RSR)	7-26
Procedure 7-15	Running Production Using Program Number Select (PNS) and UOP Production Start	7-27
Procedure 7-16	Adjusting Programs During Program or Production Run	7-30
Procedure 7-17	Using the MANUAL FCTNS Menu	7-33
Procedure 8-1	Displaying the User Screen	8-5
Procedure 8-2	Displaying and Setting Registers	8-6

Procedure 8-3	Displaying and Setting Position Registers	8-8
Procedure 8-4	Displaying the Pallet Register Screen	8-11
Procedure 8-5	Displaying and Setting System Variables	8-13
Procedure 8-6	Displaying Safety Signal Status	8-17
Procedure 8-7	Displaying the Program Timer Screen	8-18
Procedure 8-8	Displaying the System Timer Menu	8-20
Procedure 8-9	Displaying the Clock Menu	8-22
Procedure 8-10	Displaying the Version Identification Status	8-23
Procedure 8-11	Displaying Memory Status	8-26
Procedure 8-12	Displaying Position Status	8-28
Procedure 8-13	Displaying Program Execution History	8-35
Procedure 8-14	Displaying and Forcing SOP I/O	8-37
Procedure 8-15	Setting Temperature and Displaying Duty Status	8-39
Procedure 9-1	Setting Up a Port	9-7
Procedure 9-2	Using a Floppy Disk and Disk Drive	9-10
Procedure 9-3	Using the Memory Card Interface	9-12
Procedure 9-4	Setting the Default Device	9-18
Procedure 9-5	Formatting a Floppy Disk	9-19
Procedure 9-6	Operating the Floppy Cassette Adapter	9-22
Procedure 9-7	Operating a Handy File	9-24
Procedure 9-8	Selecting a Program on the Select Menu	9-28
Procedure 9-9	Saving a Program to a Disk	9-29
Procedure 9-10	Loading a Program	9-31
Procedure 9-11	Copying a Program within the SELECT Menu	9-33
Procedure 9-12	Deleting a Program from the SELECT Menu	9-34
Procedure 9-13	Printing a Program	9-36
Procedure 9-14	Printing a Teach Pendant Screen	9-38
Procedure 9-15	Generating a Directory of Files	9-42
Procedure 9-16	Loading Files Using the FILE Menu	9-45
Procedure 9-17	Restoring BACKUP Files Using the FILE Menu	9-47
Procedure 9-18	Backing Up System Files, TP Programs, and Application Files to Disk	9-55
Procedure 9-19	Displaying the Contents of a Text (ASCII) File	9-58
Procedure 9-20	Copying Files to a Disk	9-59
Procedure 9-21	Deleting Files from a Disk	9-62
Procedure 9-22	Saving Files to the Default Device	9-64
Procedure 9-23	Checking and Purging File Memory	9-65
Procedure 9-24	Backing up a Controller to a Floppy or Memory Card Device	9-68
Procedure 9-25	Restoring a Controller after a Backup	9-72
Procedure 10-1	Using Mirror Image	10-7
Procedure 10-2	Using the Shift Utility	10-21
Procedure 10-3	Setting the Conditions for the Space Check Function	10-28
Procedure 10-4	Executing Multiple Programs Using the Standard Operator Panel (SOP) CYCLE START Button	10-32
Procedure 10-5	Running Multiple Programs Using Program Number Select (PNS)	10-33
Procedure 10-6	Setting Reference Position	10-44
Procedure 10-7	Setting Up Shape Information	10-52

Procedure 10–8 Using the Shape Schedule and DETAIL Screen	10–53
Procedure 10–9 Performing Shape Adjustments Using the Shape Adjust Utility	10–74
Procedure 10–10 Setting up the Motion Group DO Output Function	10–77
Procedure 10–11 Setting Up Soft Float Schedules	10–88
Procedure 10–12 Settings for the Continuous Turn Function	10–94
Procedure 10–13 Connecting a Factory Terminal	10–100
Procedure 10–14 Starting the Factory Terminal	10–101
Procedure 10–15 Executing a Tool Change or Shift	10–115
Procedure 10–16 Executing a User Coordinate Change or Shift	10–118
Procedure 10–17 Recording a TIME BEFORE or TIME AFTER Instruction	10–123
Procedure 10–18 Creating a Condition Handler Program	10–136
Procedure 10–19 Creating an ACTION Program	10–137
Procedure 10–20 Creating a Condition Handler Program (Example)	10–137
Procedure 10–21 Starting a Condition Handler Program from a Teach Pendant Program	10–139
Procedure 10–22 Setting Up Collision Guard	10–144
Procedure 10–23 Setting Up Error Recovery Items	10–153
Procedure 10–24 Setting Up Alarm Code Monitoring	10–154
Procedure 10–25 Setting Up Digital Input Alarms	10–157
Procedure 10–26 Adding Error Recovery Instructions to a Program	10–160
Procedure 10–27 Manual Operation of Error Recovery	10–163
Procedure 10–28 Setting Up Auto Normal	10–174
Procedure 10–29 Executing Auto Normal to Perform a Search	10–175
Procedure 10–30 Performing the Cross Car Mirror Function	10–187
Procedure 10–31 Adjusting a UTOOL	10–190
Procedure 10–32 Performing the Flip Knuckle Function	10–195
Procedure 10–33 Using the Limit Set Function to Set Software Axis Limits	10–198
Procedure 12–1 Setting Up the String Sensor and TCP Attachment Device	12–3
Procedure 12–2 Replacing the Auto TCP String Sensor String	12–4
Procedure 12–3 Installing Sensor I/O Signal Cables	12–5
Procedure 12–4 Configuring the I/O Ports	12–6
Procedure 12–5 Setting Up the Sensor	12–8
Procedure 12–6 Testing Data Ports	12–10
Procedure 12–7 Setting Up TCP Orientation	12–11
Procedure 12–8 Automatically Generating Positions	12–16
Procedure 12–9 Automatically Calculating a New UTOOL TCP	12–21
Procedure A–1 Error Recovery Recommendation	A–1
Procedure A–2 Displaying the Alarm Log Screen	A–3
Procedure A–3 Recovering from an Overtravel Error	A–11
Procedure A–4 Recovering from a Hand Breakage	A–13
Procedure A–5 Using the Mastering Routine	A–14
Procedure C–1 Performing a Controlled Start	C–4
Procedure C–2 Performing a CTRL2 Start	C–6
Procedure C–3 Performing a Cold Start	C–7
Procedure C–4 Performing a Semi Hot Start	C–9
Procedure C–5 Performing a Re-Init Start using CMOSINIT	C–10
Procedure C–6 Using BootROM Utilities	C–14

Procedure C-7 Using EMON> Utilities	C-15
Procedure C-8 Using DIAG> Utilities	C-17
Procedure C-9 Using INSTALL Utilities	C-19
Procedure C-10 Using FROM Utilities	C-20
Procedure C-11 Using MCARD Utilities	C-21
Procedure E-1 Preparing the Robot for Mastering	E-3
Procedure E-2 Mastering to a Fixture (for all robots except S-6 and S-12)	E-6
Procedure E-3 Mastering to Zero Degrees (for all robots except S-420iR)	E-8
Procedure E-4 Mastering a Single Axis	E-10
Procedure E-5 Recording the Quick Master Reference Position	E-13
Procedure E-6 Quick Mastering the Robot	E-15
Procedure F-1 Eliminating a UOP Configuration	F-2
Procedure F-2 Reconfiguring Digital I/O Signals	F-2
Procedure F-3 Resetting Overtravel	F-2

List of Figures

Figure 1-1. System Overview	1-2
Figure 1-2. Major and Minor Axes	1-3
Figure 1-3. A-520i Robot Model	1-4
Figure 1-4. M-6i Robot Model	1-5
Figure 1-5. M-16i Robot Model	1-5
Figure 1-6. M-400 Robot Model	1-6
Figure 1-7. M-410i Robot Model	1-6
Figure 1-8. M-500 Robot Model	1-7
Figure 1-9. M-710i Robot Model	1-7
Figure 1-10. S-6 Robot Model	1-8
Figure 1-11. S-12 Robot Model	1-8
Figure 1-12. S-420iF Robot Model	1-9
Figure 1-13. S-420iW Robot Model	1-9
Figure 1-14. S-450S Robot Model	1-10
Figure 1-15. S-500 Robot Model	1-10
Figure 1-16. S-700 Robot Model	1-11
Figure 1-17. S-800 Robot Model	1-11
Figure 1-18. S-900 Robot Model	1-12
Figure 1-19. Finger Type Gripper	1-13
Figure 1-20. Vacuum Type Gripper	1-13
Figure 1-21. R-J2 Controllers	1-14
Figure 1-22. R-J2 Controller Capabilities	1-15
Figure 1-23. R-J2 Controller Possible Configuration	1-16
Figure 1-24. Teach Pendant	1-18
Figure 1-25. Full Menus (pages 1 and 2)	1-19
Figure 1-26. Quick Menus (pages 1 and 2)	1-19
Figure 1-27. DEADMAN Switch	1-20

Figure 1–28. R-J2 Controller Standard Operator Panels	1–21
Figure 1–29. Mode Select Switch (European Controller)	1–22
Figure 1–30. MODE SELECT Switch (Control Reliable – RS-1/RS-4 – Controller)	1–23
Figure 1–31. Effect of Opening the Safety Fence While in AUTO Mode	1–25
Figure 1–32. Program Example	1–34
Figure 1–33. Full Menus (pages 1 and 2)	1–36
Figure 1–34. FCTN Menu (pages 1 and 2)	1–36
Figure 1–35. UTILITIES Menu Map	1–37
Figure 1–36. UTILITIES Menu Map (continued)	1–38
Figure 1–37. TEST CYCLE Menu Map	1–39
Figure 1–38. MANUAL FCTNS Menu Map	1–40
Figure 1–39. ALARM Menu Map	1–40
Figure 1–40. I/O Menu Map	1–41
Figure 1–41. I/O Menu Map (Continued)	1–42
Figure 1–42. I/O Menu Map (Continued)	1–43
Figure 1–43. SETUP Menu Map	1–43
Figure 1–44. SETUP Menu Map (Continued, Page 2)	1–44
Figure 1–45. SETUP Menu Map (Continued, Page 3)	1–45
Figure 1–46. SETUP Menu Map (Continued, Page 4)	1–46
Figure 1–47. SETUP Menu Map (Continued, Page 5)	1–47
Figure 1–48. SETUP Menu Map (Continued, Page 6)	1–48
Figure 1–49. FILE Menu Map	1–49
Figure 1–50. SELECT Menu Map	1–50
Figure 1–51. EDIT Menu Map	1–51
Figure 1–52. DATA Menu Map	1–52
Figure 1–53. STATUS Menu Map	1–53
Figure 1–54. POSITION Menu Map	1–54
Figure 1–55. SYSTEM Menu Map	1–54
Figure 2–1. Jog Speed Display	2–6
Figure 2–2. Jog Speed Keys	2–6
Figure 2–3. COORD Display	2–7
Figure 2–4. JOINT Coordinate System	2–7
Figure 2–5. XYZ Coordinate System	2–8
Figure 2–6. TOOL Coordinate System	2–8
Figure 2–7. Wrist Jogging Display	2–9
Figure 2–8. Remote TCP Display	2–9
Figure 2–9. Sub-group Display	2–10
Figure 2–10. Jog Menu	2–15
Figure 3–1. Process I/O Board Hardware Layout	3–3
Figure 3–2. Modular I/O Hardware Layout	3–4
Figure 3–3. Process I/O Board Hardware Layout for Analog I/O	3–5
Figure 3–4. Modular I/O Hardware Layout For Analog I/O	3–6

Figure 3–5. Process I/O Board Hardware Layout for Digital I/O	3–11
Figure 3–6. Modular I/O Hardware Layout For Digital I/O	3–11
Figure 3–7. Process I/O Board Hardware Layout for Group I/O	3–19
Figure 3–8. Modular I/O Hardware Layout For Group I/O	3–20
Figure 3–9. Process I/O Board Hardware Layout	3–34
Figure 3–10. Modular I/O Hardware Layout	3–34
Figure 3–11. RSR Timing Diagram	3–37
Figure 3–12. PNS Timing Diagram	3–38
Figure 3–13. Model B I/O – i-size Controller	3–43
Figure 3–14. Model B I/O – B-size Controller	3–44
Figure 3–15. Example Distributed I/O Setup Block Diagram	3–45
Figure 3–16. Interface Unit DIP Switches	3–46
Figure 3–17. Basic Digital I/O Module DIP Switches	3–47
Figure 3–18. PLC I/O Process I/O Board Hardware Layout	3–64
Figure 3–19. PLC I/O Modular I/O Hardware Layout	3–64
Figure 3–20. World Frame	3–78
Figure 3–21. Moving a Frame	3–79
Figure 3–22. Tool Frame	3–80
Figure 3–23. Defining the Orientation of the Origin	3–87
Figure 3–24. World and User Frames	3–94
Figure 3–25. Defining the Origin	3–97
Figure 3–26. Defining the X Direction Point	3–97
Figure 3–27. Defining the X-Y Plane	3–98
Figure 3–28. Defining the Origin	3–101
Figure 3–29. Defining the X Direction Point	3–102
Figure 3–30. Defining the X-Y Plane	3–102
Figure 3–31. Defining the Second Origin	3–103
Figure 3–32. Remote TCP Frame	3–110
Figure 3–33. Touching the TCP of the Robot Tool to the Remote TCP	3–113
Figure 3–34. Touching the TCP of the Robot Tool to the Remote TCP	3–115
Figure 3–36. Defining the Origin	3–122
Figure 3–37. Defining the X Direction Point	3–123
Figure 3–38. Defining the X-Y Plane	3–123
Figure 3–39. Teach Pendant User Keys	3–137
Figure 3–40. Teach Pendant User Keys	3–142
Figure 3–41. Inertia Equations	3–183
Figure 5–1. Continuous Termination Type for Palletizing	5–2
Figure 5–2. Home Position	5–6
Figure 5–3. Repair Position	5–7
Figure 5–4. Safe Position	5–8
Figure 5–5. Writing and Modifying a Program	5–9
Figure 5–6. Background Edit Process	5–31

Figure 5–7. Background Edit Process (continued)	5–32
Figure 6–1. Program Example	6–3
Figure 6–2. Motion Instruction Example	6–11
Figure 6–3. Joint Motion Type	6–12
Figure 6–4. Linear Motion Type	6–13
Figure 6–5. Linear Motion Type Used to Rotate About the Tool Center Point	6–14
Figure 6–6. Circular Motion Type	6–15
Figure 6–7. Restart of Circular Motion Instruction	6–16
Figure 6–8. Restart of Circular Motion Instruction	6–16
Figure 6–9. Effect of Via Point Location	6–17
Figure 6–10. Effect of UTOOL Accuracy	6–18
Figure 6–11. Positional Information	6–20
Figure 6–12. Frame Number of Positional Data Example Program	6–22
Figure 6–13. Example of the Sec Speed Feature	6–23
Figure 6–14. Variable Motion Speed Program Execution Example	6–24
Figure 6–15. Robot Motion with Fine Termination Type	6–32
Figure 6–16. Robot Motion with Continuous Termination Type	6–33
Figure 6–17. Acceleration Override	6–35
Figure 6–18. SKIP LBL[x] Motion Option Example	6–36
Figure 6–19. Tool Offset Instruction	6–40
Figure 6–20. Position Representation Screen	6–40
Figure 6–21. Remote TCP Motion Option	6–42
Figure 6–22. Remote TCP (RTCP) Motion Option Example	6–43
Figure 6–23. Corner Rounding	6–44
Figure 6–24. Pick and Place	6–46
Figure 6–25. The Effect of Corner Distance on Corner Rounding	6–49
Figure 6–26. Half Distance Rule	6–52
Figure 6–27. Short Segment Path WITHOUT AccuPath	6–53
Figure 6–28. Short Segment Path with AccuPath	6–53
Figure 6–29. Path Orientation	6–54
Figure 6–30. Teaching a Small Corner	6–55
Figure 6–31. Teaching a Flexible Path	6–56
Figure 6–32. Palletizing Program Example	6–61
Figure 6–33. Example of PALLETIZING-B Instruction	6–62
Figure 6–34. Example of PALLETIZING-BX Instruction	6–68
Figure 6–35. Example of PALLETIZING-E Instruction	6–75
Figure 6–36. Example of PALLETIZING-EX Instruction	6–83
Figure 6–37. Direct and Indirect Addressing Example	6–91
Figure 6–38. PL[x] = [value]	6–92
Figure 6–39. PL[x] = [value] [operator] [value]	6–92
Figure 6–40. Direct and Indirect Addressing Example	6–93
Figure 6–41. R[x] = [value]	6–94

Figure 6-42. R[x] = [value] [operator] [value]	6-95
Figure 6-43. PR[GRPn:x] = [value]	6-96
Figure 6-44. PR[GRPn:x] = [value] [operator] [value]	6-97
Figure 6-45. Position Register Element PR[i,j]	6-97
Figure 6-46. PR[i,j] = [value]	6-98
Figure 6-47. PR[i,j] = [value] [operator] [value]	6-99
Figure 6-48. R[x] = DI[x]	6-100
Figure 6-49. DO[x] = ON/OFF	6-100
Figure 6-50. DO[x] = PULSE [,width]	6-101
Figure 6-51. DO[x] = R[x]	6-101
Figure 6-52. R[x] = RI[x]	6-101
Figure 6-53. RO[x] = ON/OFF	6-102
Figure 6-54. RO[x] = PULSE [,width]	6-102
Figure 6-55. RO[x] = R[x]	6-102
Figure 6-56. R[x] = AI[x]	6-103
Figure 6-57. AO[x] = value	6-103
Figure 6-58. R[x] = GI[x]	6-104
Figure 6-59. GO[x] = value	6-104
Figure 6-60. LBL[x]	6-105
Figure 6-61. JMP LBL[x]	6-105
Figure 6-62. CALL Program Instruction	6-106
Figure 6-63. Program End Instruction	6-106
Figure 6-64. Register IF Instruction	6-106
Figure 6-65. I/O IF Instruction for DI/DO, RI/RO, SI/SO and UI/UO	6-107
Figure 6-66. I/O IF Instruction for PL	6-107
Figure 6-67. I/O IF Instruction for R, AI/AO, GI/GO and System Variable	6-107
Figure 6-68. Select Instruction	6-108
Figure 6-69. Wait Time	6-109
Figure 6-70. WAIT Condition	6-110
Figure 6-71. WAIT Condition	6-110
Figure 6-72. WAIT Condition	6-110
Figure 6-73. RSR Enable/Disable	6-112
Figure 6-74. User Alarm	6-112
Figure 6-75. Timer	6-113
Figure 6-76. OVERRIDE	6-113
Figure 6-77. Message Instruction	6-113
Figure 6-78. Parameter Name Write Instruction	6-114
Figure 6-79. Parameter Name Read Instruction	6-115
Figure 6-80. JOINT_MAX_SPEED Instruction – Multiple Motion Group Syntax	6-118
Figure 6-81. LINEAR_MAX_SPEED Instruction – Multiple Motion Group Syntax	6-118
Figure 6-82. JOINT_MAX_SPEED Instruction – Single Motion Group Syntax	6-118
Figure 6-83. LINEAR_MAX_SPEED Instruction – Single Motion Group Syntax	6-118

Figure 6–84. Skip Condition	6–119
Figure 6–85. Skip Condition	6–119
Figure 6–86. Skip Condition	6–119
Figure 6–87. Offset Condition	6–121
Figure 6–88. UFRAME_NUM=[value]	6–122
Figure 6–89. UTOOL_NUM=[value]	6–122
Figure 6–90. UFRAME[i] = PR[x]	6–123
Figure 6–91. UTOOL[i] = PR[x]	6–123
Figure 6–92. Tool Offset Condition Instruction	6–124
Figure 6–93. SEMAPHORE[i] = ON/OFF	6–125
Figure 6–94. WAIT SEMAPHORE[x] [time]	6–125
Figure 6–95. RUN program	6–125
Figure 6–96. Macro Command Instruction	6–126
Figure 6–97. PAUSE	6–127
Figure 6–98. ABORT	6–127
Figure 6–99. Error Program	6–128
Figure 6–100. RESUME_PROG = program	6–128
Figure 6–101. SEND R[x]	6–129
Figure 6–102. RCV R[x] LBL[x]	6–129
Figure 6–103. RCV R[x], LBL[x]	6–130
Figure 6–104. CALMATRIX Instruction	6–130
Figure 6–105. LOCK PREG Instruction	6–134
Figure 6–106. UNLOCK PREG Instruction	6–134
Figure 6–107. MONITOR Instruction	6–135
Figure 6–108. MONITOR END Instruction	6–135
Figure 6–109. Condition for Register, System Variable, and I/O Parameters	6–136
Figure 6–110. Condition2 for I/O	6–136
Figure 6–111. Condition for Error Status	6–136
Figure 6–112. Payload Instruction	6–137
Figure 6–113. Example of Using PAYLOAD[GPx:y] Instructions in a Teach Pendant Program ..	6–137
Figure 6–114. Inertia Equations	6–138
Figure 6–115. Example of Enabling and Disabling Collision Guard in a Teach Pendant Program	6–139
Figure 7–1. Resume Tolerance Example	7–5
Figure 7–2. Example Program Showing Backward Execution	7–13
Figure 8–1. Teach Pendant Status Indicators	8–2
Figure 8–2. R-J2 Controller (i-Size) Standard Operator Panel	8–3
Figure 8–3. R-J2 Controller (B-Size) Standard Operator Panel	8–4
Figure 8–4. Turn Number and Joint Placement Display on Position Screen	8–30
Figure 8–5. Turn Number Display Configuration	8–31
Figure 8–6. Joint Placement Configuration Examples for Fully Articulated Robots	8–32
Figure 8–7. Joint Placement Configuration Examples for Horizontally Articulated Robots	8–33
Figure 8–8. \$SCR_GRP[group].\$turn_axis[i] for Turn Number Display Configuration	8–33

Figure 8–9. Duty Diagnosis Screen	8–38
Figure 9–1. Location of Standard and Optional Ports on an i-Size Controller	9–4
Figure 9–2. PS-100 or PS-110 Connected to the i-Size and B-Size Controller	9–8
Figure 9–3. PS-200 Connected to the i-Size and B-Size Controller	9–8
Figure 9–4. Handy File Connected to the Controller	9–9
Figure 9–5. FLOPPY CASSETTE ADAPTER Connected to the Controller	9–9
Figure 9–6. Memory Card Interface and Memory Card Connected to the i-size Controller	9–11
Figure 9–7. Memory Card Interface and Memory Card Connected to the B-Size Controller	9–12
Figure 9–8. R-J2 Controller Disconnect Handle and Latch	9–13
Figure 9–9. Memory Card Interface Location on an i-size Controller	9–14
Figure 9–10. Memory Card Interface Location on a B-Size Controller	9–14
Figure 9–11. Inserting a Memory Card with the ER-2 Printed Circuit Board	9–15
Figure 9–12. Inserting a Memory Card without an ER-2 Printed Circuit Board	9–16
Figure 9–13. Floppy Cassette Adapter	9–21
Figure 9–14. Status indicator LEDs	9–21
Figure 9–15. Handy File	9–23
Figure 10–1. Parallel Mirror Image with Mirror Plane in Center of Robot	10–4
Figure 10–2. Parallel Mirror Image with Mirror Plane Offset from Center of Robot	10–5
Figure 10–3. Rotational Mirror Image	10–5
Figure 10–4. Example of Robot Axes Only Shift	10–6
Figure 10–5. Example of Extended Axes Integrated Shift	10–6
Figure 10–6. Example of With Extended Axes Shift	10–7
Figure 10–7. Mirroring an Entire Program	10–9
Figure 10–8. Mirroring a Portion of a Program	10–10
Figure 10–9. Shifting an Entire Program	10–16
Figure 10–10. Shifting Portions of a Program	10–17
Figure 10–11. Parallel Shift	10–17
Figure 10–12. Parallel and Rotating Shift	10–18
Figure 10–13. Example of Robot Axes Only Shift	10–19
Figure 10–14. Example of Extended Axes Integrated Shift	10–19
Figure 10–15. Example of With Extended Axes Shift	10–20
Figure 10–16. Example of With Extended Axes Only Shift	10–20
Figure 10–17. Example of a Replace Extended Axes Shift	10–21
Figure 10–18. Turn Numbers	10–25
Figure 10–19. Using Register Instructions to Synchronize Program Execution	10–31
Figure 10–20. Multi-Tasking Using the RUN Program Instruction	10–34
Figure 10–21. Single Step Execution Example	10–35
Figure 10–22. Single Step Backward Execution	10–36
Figure 10–23. Backward Execution of a RUN Instruction Example	10–36
Figure 10–24. Two Methods of Angle Entry Shift Function	10–38
Figure 10–25. Angle Entry Shift Screen Structure	10–39
Figure 10–26. Position Register Look-Ahead Program Example	10–48

Figure 10–27. Circle Schedule SCHEDULE Screen	10–55
Figure 10–28. Circle Schedule DETAIL Screen	10–55
Figure 10–29. Circle Schedule Terminology	10–57
Figure 10–30. Hexagon Schedule SCHEDULE Screen	10–57
Figure 10–31. Hexagon Schedule DETAIL Screen	10–57
Figure 10–32. Hexagon Schedule Terminology	10–59
Figure 10–33. Rectangle Schedule SCHEDULE Screen	10–60
Figure 10–34. Rectangle Schedule DETAIL Screen	10–60
Figure 10–35. Rectangle Schedule Terminology	10–62
Figure 10–36. Slot Schedule SCHEDULE Screen	10–62
Figure 10–37. Slot Schedule DETAIL Screen	10–63
Figure 10–38. Slot Schedule Terminology	10–64
Figure 10–39. Shape Generation Program Example	10–65
Figure 10–40. Example of Using Override Registers	10–69
Figure 10–41. Teach and Production Modes	10–70
Figure 10–42. UTILITIES Shape Adjust SCHEDULE Screen	10–71
Figure 10–43. UTILITIES Shape Adjust DETAIL Screen	10–71
Figure 10–44. Shape Adjust Utility: X and Y Offsets	10–72
Figure 10–45. UTILITIES Shape Adjust: Shape Rotation	10–72
Figure 10–46. Shape Rotation Program Example	10–73
Figure 10–47. Shape Adjust Utility: Offset and Rotated Shapes	10–73
Figure 10–48. Shape Frames	10–75
Figure 10–49. Motion Group DO Output Function Program Examples	10–78
Figure 10–50. Motion Start Delay Memory Buffers	10–81
Figure 10–51. SOFTFLOAT[n] Independent Instruction Example	10–90
Figure 10–52. SOFTFLOAT[n] Motion Option Example	10–91
Figure 10–53. Continuous Turn Example	10–92
Figure 10–54. Shortest Rotational Distance Rule	10–93
Figure 10–55. Factory Terminal	10–99
Figure 10–56. Palletizing Example	10–107
Figure 10–57. Coordinates Offset Screens	10–110
Figure 10–58. TCP Fixed Method	10–112
Figure 10–59. Robot Fixed Method	10–113
Figure 10–60. Robot Fixed Method	10–113
Figure 10–61. TIME BEFORE / TIME AFTER Motion Option Instructions	10–121
Figure 10–62. Timing Sequence (TIME BEFORE instruction)	10–122
Figure 10–63. Timing Sequence (AFTER instruction)	10–122
Figure 10–64. Timing Sequence(TIME BEFORE instruction)	10–122
Figure 10–65. Main and Sub Program Examples	10–125
Figure 10–66. Program example for TIME BEFORE instruction	10–125
Figure 10–67. Condition Monitor Function	10–127
Figure 10–68. Sample, Condition Handler, and Action Programs	10–127

Figure 10–69. Condition for Register, System Variable, and I/O Parameters	10–131
Figure 10–70. Condition2 for I/O	10–131
Figure 10–71. Condition for Error status	10–131
Figure 10–72. Program Monitor Menu	10–132
Figure 10–73. System Monitor Menu	10–133
Figure 10–74. Collision Guard Adjust Macro Program	10–142
Figure 10–75. Example of Enabling and Disabling Collision Guard in a Teach Pendant Program	10–145
Figure 10–76. Resume Program Example	10–147
Figure 10–77. Maintenance Program Example	10–148
Figure 10–78. Error Recovery Setup Screen	10–152
Figure 10–79. Setting User Alarm Screen	10–156
Figure 10–80. RESUME_PROGRAM Instruction	10–158
Figure 10–81. CLEAR_RESUME_PROG Instruction	10–158
Figure 10–82. WELD.TP Example Program	10–158
Figure 10–83. WIRE_CUT.TP (Resume Program) Example Program	10–158
Figure 10–84. MAINT_PROGRAM Instruction	10–159
Figure 10–85. RETURN_PATH_DSBL Instruction	10–159
Figure 10–86. WELD.TP Example Program	10–159
Figure 10–87. Normal Operation Auto Start Mode	10–164
Figure 10–88. Normal Operation without Execution of Resume Program	10–165
Figure 10–89. Resume Program Aborted	10–166
Figure 10–90. Normal Operation (Automatic Start DISABLED)	10–167
Figure 10–91. Auto Mode When an Undefined Alarm Occurs	10–168
Figure 10–92. Laser Height Sensor	10–170
Figure 10–93. Search Motion	10–171
Figure 10–94. Positions and Frames Defined After the Search	10–172
Figure 10–95. Shape Frames	10–172
Figure 10–96. Cross Car Mirror Function	10–185
Figure 10–97. New UTOOL and UFRAME Values Displayed in Mirrored Destination Program .	10–186
Figure 10–98. Flip Knuckle Function	10–194
Figure 10–99. Top view of Robot Base for Locating Axis 1 Hard Stop Location	10–200
Figure 11–1. Composition of FANUC Sensor Data Packet	11–2
Figure 11–2. Handshaking When the Contents of a Register is Sent to the Sensor	11–4
Figure 11–3. Handshaking When a Register Number and Its Data is Received	11–4
Figure 11–4. Handshaking When Position Register Data is Received	11–5
Figure 11–5. Handshaking When Three Point Data is Received	11–5
Figure 11–6. Handshaking When the Transformation Matrix is Received	11–5
Figure 11–7. FANUC Sensor Interface Program Example	11–6
Figure 12–1. Auto TCP String Sensor and TCP Attachment Device	12–3
Figure 12–2. Auto TCP String Sensor	12–4
Figure 12–3. String Sensor I/O Signal Cables	12–5
Figure 12–4. Auto TCP Point Generation	12–14

Figure A-1. Hexadecimal Error Message Display	A-10
Figure B-1. Built-in and Remote CRT/KBs	B-1
Figure D-1. /PROG PREG_ELE	D-2
Figure D-2. /PROG PREG_VAL	D-2
Figure D-3. /PROG REG_AI	D-3
Figure D-4. /PROG REG_GI	D-3
Figure D-5. /PROGRAM CYCLECNT – REGISTER INCREMENT	D-4
Figure D-6. /PROGRAM SIGNAL – Group Output; WAIT and PULSE Instruction	D-4
Figure D-7. /PROGRAM MAIN – LABELS	D-5
Figure D-8. /PROGRAM MAIN – LABEL; JUMP LABEL MESSAGE	D-5
Figure D-9. /PROG MAIN	D-6
Figure F-1. Transportation	F-1

List of Tables

Table 1-1. Robot Stop Variation for European Controllers	1-27
Table 1-2. Robot Servo Status for Control Reliable (RS-1/RS-4) Option	1-28
Table 2-1. Relationship of Jogging to \$SCR.\$TPMOTNENABL	2-5
Table 2-2. Jog Speed Values	2-6
Table 2-3. Sub-Group Example	2-10
Table 3-1. I/O Interconnect Screen Items	3-29
Table 3-2. Relationship Between the MODE SELECT Switch Signals and Modes of Operation	3-31
Table 3-3. UOP Input Signals	3-35
Table 3-4. UOP Output Signals	3-39
Table 3-5. Communication Speed Settings for Switches Q and H	3-46
Table 3-6. Unit Number Settings of Switches 16, 8, 4, 2, and 1	3-48
Table 3-7. Rack Assignments for Different Kinds of I/O	3-50
Table 3-8. Slot Assignments for Different Kinds of I/O	3-51
Table 3-9. Rack Assignments for Different Kinds of I/O	3-58
Table 3-10. Slot Assignments for Different Kinds of I/O	3-59
Table 3-11. Device Names	3-70
Table 3-12. Devices that have Access to the Detail Screen	3-71
Table 3-13. RSR Setup Item Description	3-132
Table 3-14. PNS Setup Item Description	3-135
Table 3-15. Brake On Hold Settings	3-151
Table 3-16. \$UALRM_SEV[n] Severity Values	3-155
Table 3-17. \$UALRM_SEV[n] Severity Values	3-157
Table 3-18. Override Select Menu Listing	3-159
Table 3-19. Password Levels	3-161
Table 3-20. Password Error Messages	3-170
Table 3-21. Password Level Screen Permissions	3-173
Table 3-22. SYSTEM Payload Screen	3-177
Table 4-1. System Configuration Setup Screen Items	4-2

Table 5-1.	Paste Methods	5-18
Table 5-2.	Troubleshoot Background Edit – Problem Cause and Remedy	5-35
Table 6-1.	\$FRM_CHKTYP Values	6-21
Table 6-2.	Example Program Operation	6-22
Table 6-3.	Range of Register Values for Specifying Variable Motion Speed	6-25
Table 6-4.	Correspondence between \$GROUP System Variables and the Teach Pendant Motion Instruction	6-57
Table 6-5.	PALLETIZING-B Pallet Editor Items	6-62
Table 6-6.	PALLETIZING-BX Pallet Editor Items	6-69
Table 6-7.	PALLETIZING-E Pallet Editor Items	6-76
Table 6-8.	PALLETIZING-BX Pallet Editor Items	6-83
Table 7-1.	Tolerance Setup Items	7-5
Table 7-2.	Test Cycle Conditions	7-10
Table 8-1.	Teach Pendant Status Indicators	8-2
Table 8-2.	Standard Operator Panel Status Indicators	8-4
Table 8-3.	Safety Signals	8-15
Table 8-4.	Program Timer Listing Screen	8-18
Table 8-5.	System Timer Menu Listing	8-20
Table 8-6.	System Timer Menu Listing	8-22
Table 8-7.	Version Identification Status Items	8-23
Table 8-8.	Memory Status	8-26
Table 8-9.	Execution History Screen	8-34
Table 8-10.	Standard Operator Panel Input Signals	8-36
Table 8-11.	Standard Operator Panel Output Signals	8-36
Table 8-12.	Duty Diagnosis Screen Items	8-39
Table 9-1.	Ports, P1 – P4	9-5
Table 9-2.	Default Communications Settings for Devices	9-5
Table 9-3.	Pin Configuration of the P3 Port DB-25 Connector	9-6
Table 9-4.	Pin Configuration of the P4 Port JD-17 Connector	9-6
Table 9-5.	Port Setting on Floppy Cassette Adapter	9-21
Table 9-6.	Status Indicator LEDs and Switches	9-22
Table 9-7.	Port Setting for Handy File	9-23
Table 9-8.	File Output Using PRINT	9-35
Table 9-9.	Types of Files	9-40
Table 9-10.	Error Log Files	9-54
Table 9-11.	Valid SAVE Function Screens	9-64
Table 10-1.	Interlock Output Signal Operation	10-26
Table 10-2.	Space Check Function Screen Items	10-27
Table 10-3.	Reference Position LISTING Screen Items	10-43
Table 10-4.	Reference Position DETAIL Screen Items	10-44
Table 10-5.	Shape Setup Information	10-51
Table 10-6.	Circle Schedule Data	10-55

Table 10-7.	Hexagon Schedule Data	10-58
Table 10-8.	Rectangle Schedule Data	10-60
Table 10-9.	Slot Schedule Data	10-63
Table 10-10.	Shape Generation Program Example Description	10-66
Table 10-11.	Calculate Shape Macros	10-67
Table 10-12.	Shape Macros	10-67
Table 10-13.	Soft Float Schedule Setup Items	10-86
Table 10-14.	Continuous Turn Setup Items	10-93
Table 10-15.	Operations on the Teach Pendant and the Corresponding Operations on the Factory Terminal	10-103
Table 10-16.	Tool Offset Screen Items	10-114
Table 10-17.	User Frame Offset Screen Items	10-117
Table 10-18.	State of Condition Monitoring	10-129
Table 10-19.	Program Monitor Menu Items	10-132
Table 10-20.	System Monitor Menu Items	10-133
Table 10-21.	Collision Guard Setup Items	10-143
Table 10-22.	Error Recovery Features	10-149
Table 10-23.	Error Recovery Setup Items	10-152
Table 10-24.	Auto Error Recovery Manual Function Screen Items	10-161
Table 10-25.	Auto Error Recovery Manual Function Detail Screen Items	10-162
Table 10-26.	Auto Normal Setup Items	10-173
Table 10-27.	Cross Car Mirror Setup Items	10-186
Table 10-28.	UTOOL Adjust Setup Items	10-189
Table 10-29.	Flip Knuckle Setup Items	10-195
Table 11-1.	FANUC Sensor Data Formats	11-3
Table 11-2.	NULL Modem Cable Pin Connector Layout	11-7
Table 12-1.	Sensor Setup Screen	12-7
Table 12-2.	Orientation Setup Menu Items	12-11
Table 12-3.	Auto Setup and Record Data Menu Items.	12-15
Table 12-4.	Auto TCP Menu Items	12-20
Table A-1.	Start Methods	A-2
Table A-2.	Error Facility Codes	A-6
Table A-3.	Effects of Error Severity	A-8
Table A-4.	Hexadecimal Notation and Axis in Error Examples	A-10
Table B-1.	Port Settings for the Factory Terminal	B-2
Table B-2.	Correspondence Between Teach Pendant and CRT/KB Keys	B-2
Table C-1.	Controlled Start Options	C-3
Table C-2.	BootROM Utilities	C-12
Table C-3.	BootROM Extended Monitor Utilities	C-15
Table C-4.	BootROM Diagnostic Utilities	C-16
Table C-5.	INSTALL Utilities	C-18
Table C-6.	Flash ROM Items	C-20
Table C-7.	Memory Card Items	C-21

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

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

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

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

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

CONSIDERING SAFETY FOR YOUR ROBOT INSTALLATION

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

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

Keeping People and Equipment Safe

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

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

Using Safety Enhancing Devices

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

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

Setting Up a Safe Workcell

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

- Sponsor your personnel for training in approved FANUC Robotics training course(s) related to your application. Never permit untrained personnel to operate the robots.
- Install a lockout device that uses an access code to prevent unauthorized persons from operating the robot.
- Use anti-tie-down logic to prevent the operator from bypassing safety measures.
- Arrange the workcell so the operator faces the workcell and can see what is going on inside the cell.

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

Staying Safe While Teaching or Manually Operating the Robot

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

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

WARNING

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

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

WARNING

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

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

Staying Safe During Automatic Operation

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

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

Staying Safe During Inspection

When inspecting the robot, be sure to

- Turn off power at the controller.
- Lock out and tag out the power source at the controller according to the policies of your plant.
- Turn off the compressed air source and relieve the air pressure.
- If robot motion is not needed for inspecting the electrical circuits, press the EMERGENCY STOP button on the operator panel.
- Never wear watches, rings, neckties, scarves, or loose clothing that could get caught in moving machinery.

Staying Safe During Maintenance

- If power is needed to check the robot motion or electrical circuits, be prepared to press the EMERGENCY STOP button, in an emergency.
- Be aware that when you remove a servomotor or brake, the associated robot arm will fall if it is not supported or resting on a hard stop. Support the arm on a solid support before you release the brake.

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

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



WARNING

Lethal voltage is present in the controller WHENEVER IT IS CONNECTED to a power source. Be extremely careful to avoid electrical shock.

HIGH VOLTAGE IS PRESENT at the input side whenever the controller is connected to a power source. Turning the disconnect or circuit breaker to the OFF position removes power from the output side of the device only.

- Release or block all stored energy. Before working on the pneumatic system, shut off the system air supply and purge the air lines.

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

**WARNING**

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

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

KEEPING MACHINE TOOLS AND EXTERNAL DEVICES SAFE

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

Programming Safety Precautions

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

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

Mechanical Safety Precautions

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

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

KEEPING THE ROBOT SAFE

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

Operating Safety Precautions

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

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

Programming Safety Precautions

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

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

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

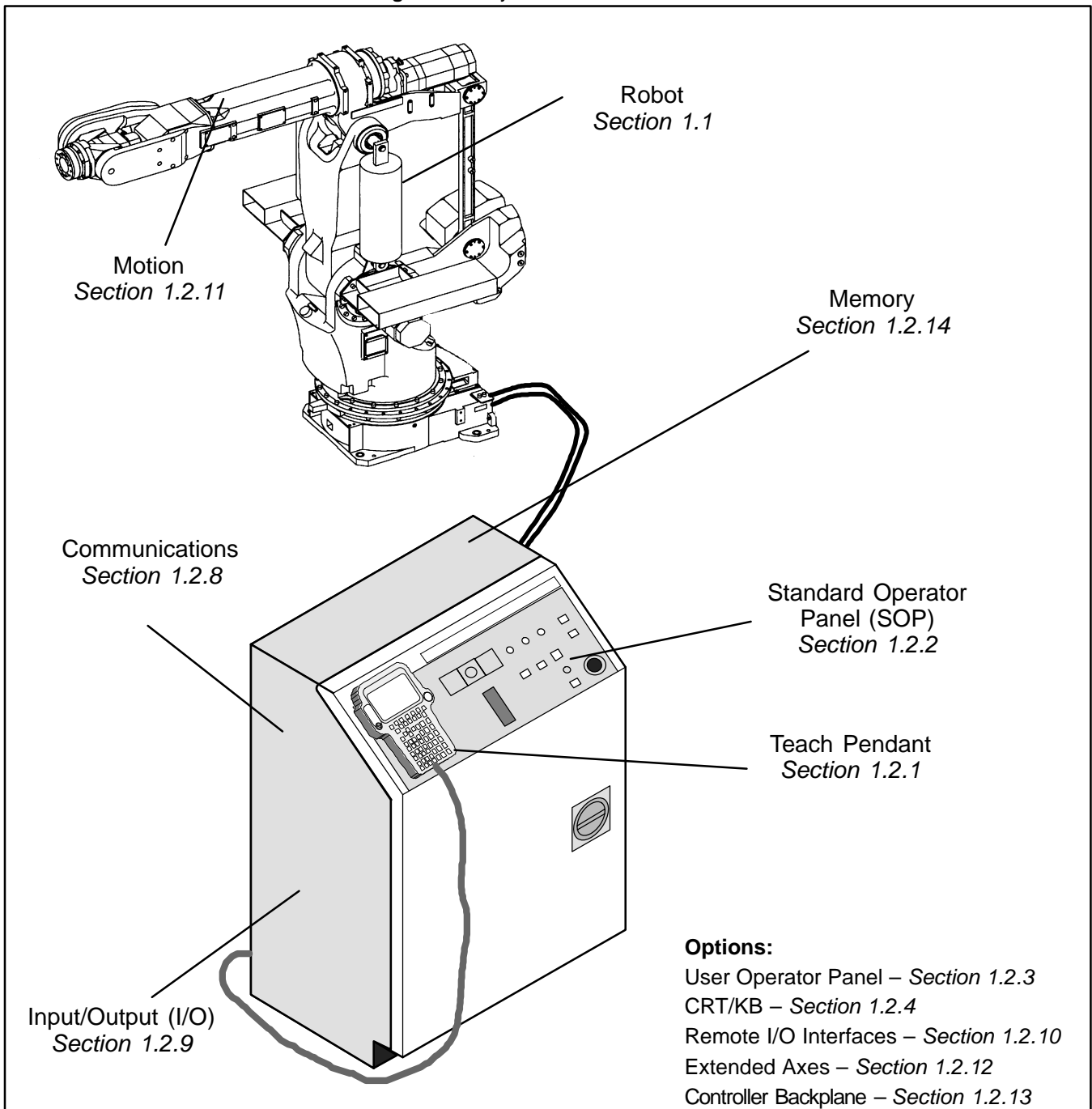
1 OVERVIEW

1 OVERVIEW

Topics In This Chapter		Page
Robot	The robot is the mechanical unit that, along with the <i>end-of-arm tooling</i> (E.O.A.T.) or gripper, actually performs the task to be completed. FANUC Robotics provides the M-410i, which is suited for palletizing applications.	1-3
	• Robot Models	1-4
	• End-of-Arm Tooling	1-13
	• Extended Axes	1-13
Controller	The R-J2 controller contains the computer that operates the robot. It houses the HandlingTool application software, controls the teach pendant and operator panel devices, and provides the necessary connections to other external devices.. . . .	1-14
	• Teach Pendant	1-17
	• Standard Operator Panel (SOP)	1-20
	• User Operator Panel (UOP)	1-26
	• CRT/KB	1-26
	• Emergency Stop Devices	1-26
	• Robot Stop Variation (European Controllers)	1-27
	• Robot Stop Variation (Control Reliable (RS-1/RS-4) option)	1-28
	• Communications	1-29
	• Input/Output (I/O)	1-30
	• Remote I/O Interfaces	1-30
	• Motion	1-31
	• Extended Axes	1-31
	• Controller Backplane	1-32
• Memory	1-32	
HandlingTool Software	HandlingTool is a software product that runs on the R-J2 controller. It uses a teach pendant interface that provides the necessary commands and menus for you to complete your task. The HandlingTool software contains all the commands and tools that allow you to communicate with the robot and external devices. These devices can include cell controllers and remote operator panels.	
	The HandlingTool software also controls all robot motion of standard axes and extended axes as well as the input/output (I/O) that is used between the controller and other devices. These other devices can include cell controllers, external file storage devices, and vision sensors.. . . .	1-33
	• Set Up	1-33
	• Program	1-33
	• Test Program	1-35
	• Run Production	1-35
Menu Maps	Menu maps illustrate how to display each teach pendant screen. Use the menu maps to look at the overall structure of the software and to locate a specific screen.. . . .	1-36
	• UTILITIES Menu Map	1-37
	• TEST CYCLE Menu Map	1-39
	• MANUAL FCTNS Menu Map	1-40
	• ALARM Menu Map	1-40
	• I/O Menu Map	1-41
	• SETUP Menu Map	1-43
	• FILE Menu Map	1-49
	• USER Menu Map	1-50
	• SELECT Menu Map	1-50
	• EDIT Menu Map	1-51
	• DATA Menu Map	1-52
	• STATUS Menu Map	1-53
• POSITION Menu Map	1-54	
• SYSTEM Menu Map	1-54	

The SYSTEM R-J2 Robot System consists of HandlingTool software, FANUC Robots and the SYSTEM R-J2 controller, referred to as the R-J2 controller or controller. The R-J2 robot system provides you with the total solution for all your robotic needs.

Figure 1-1. System Overview

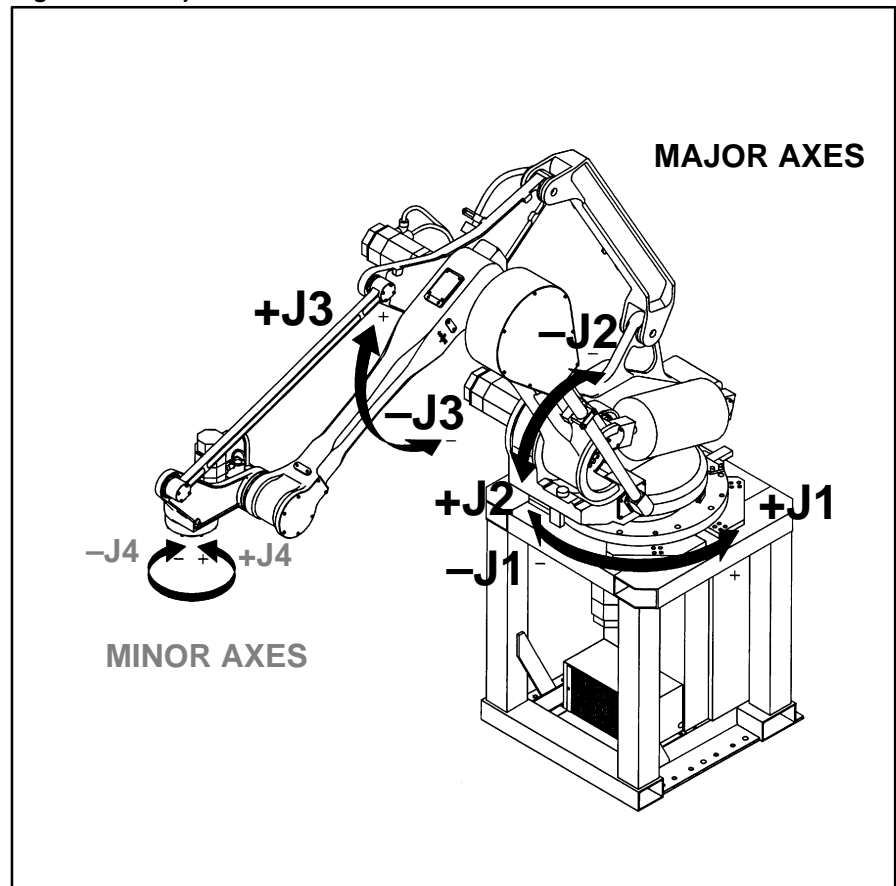


1.1 ROBOT

A robot is a series of mechanical links driven by servomotors. The area at each junction between the links is a joint, or axis. The first three axes make up the major axes as shown in Figure 1-2. A robot is classified by the number of linear and rotational major axes.

The major and minor axes shown in Figure 1-2 move the tooling at the end of the robot arm. The movements are twisting, up-and-down, and side-to-side motions.

Figure 1-2. Major and Minor Axes



1.1.1 Robot Models

The FANUC Robotics robot models used for material handling applications are:

- FANUC Robot A-520i
- FANUC Robot C-100
- FANUC Robot F-100
- FANUC Robot M-6i
- FANUC Robot M-16i
- FANUC Robot M-400
- FANUC Robot M-410i/iW/iWX
- FANUC Robot M-500
- FANUC Robot M-710i
- FANUC Robot S-6
- FANUC Robot S-12
- FANUC Robot S-420A
- FANUC Robot S-420iF/iF2.85/iL/iR/iSiW/iW2.85/iW175kg
- FANUC Robot S-450
- FANUC Robot S-500
- FANUC Robot S-700
- FANUC Robot S-800
- FANUC Robot S-900
- FANUC Robot Coordinated robot
- FANUC Robot Nobot

Figure 1-3 through Figure 1-18 display each of these robot models.

Figure 1-3. A-520i Robot Model

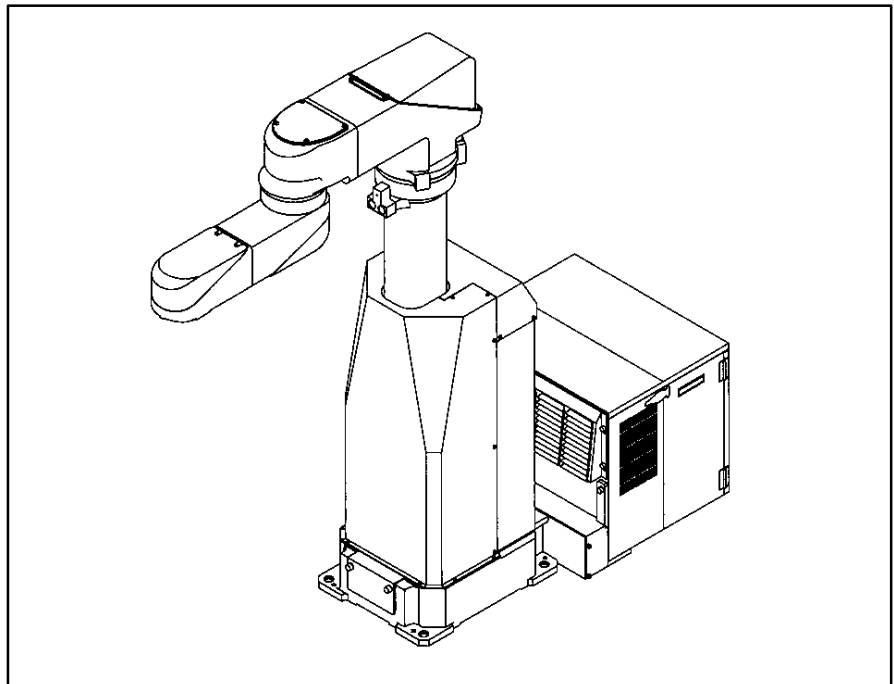


Figure 1-4. M-6i Robot Model

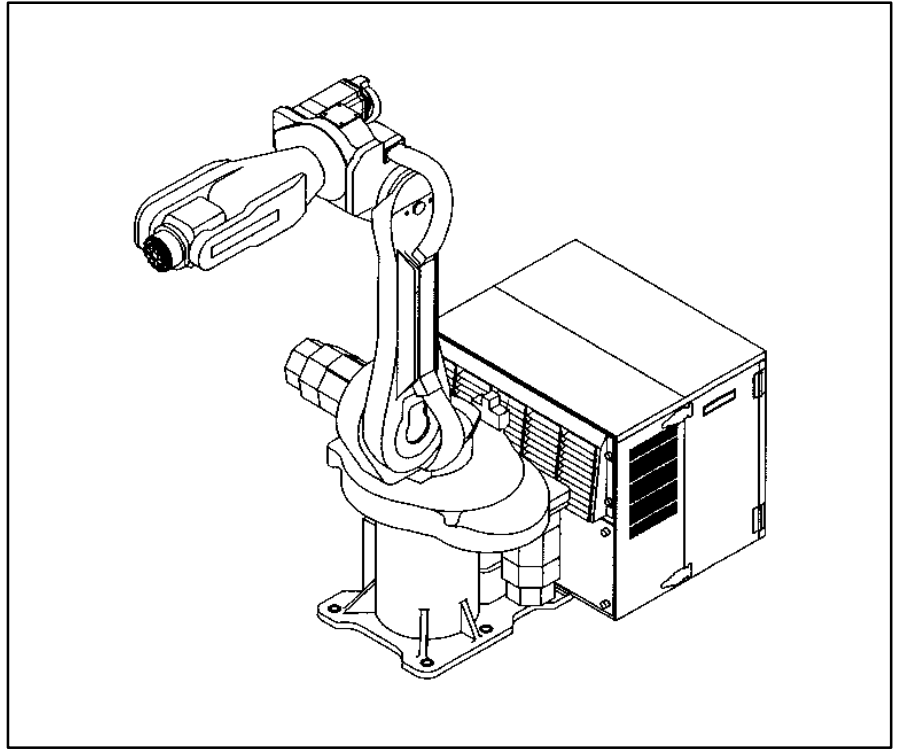


Figure 1-5. M-16i Robot Model

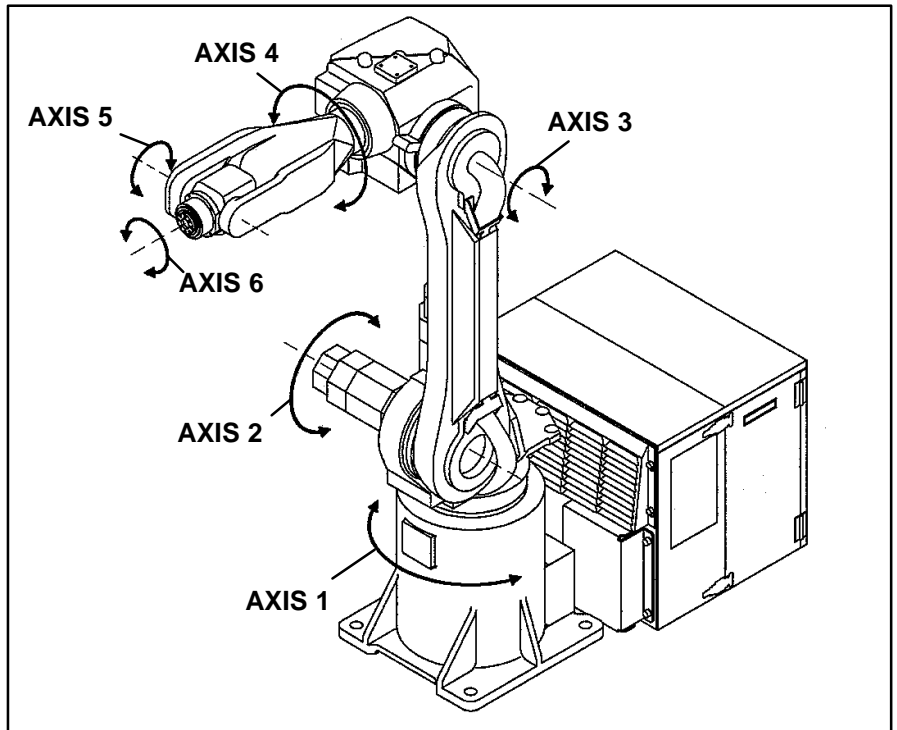


Figure 1-6. M-400 Robot Model

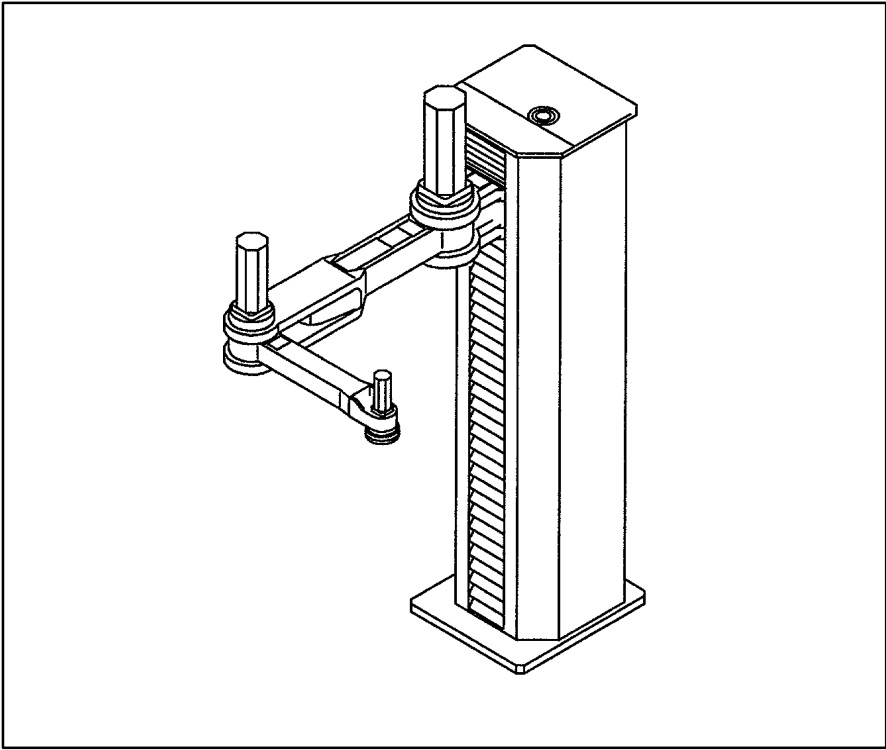


Figure 1-7. M-410i Robot Model

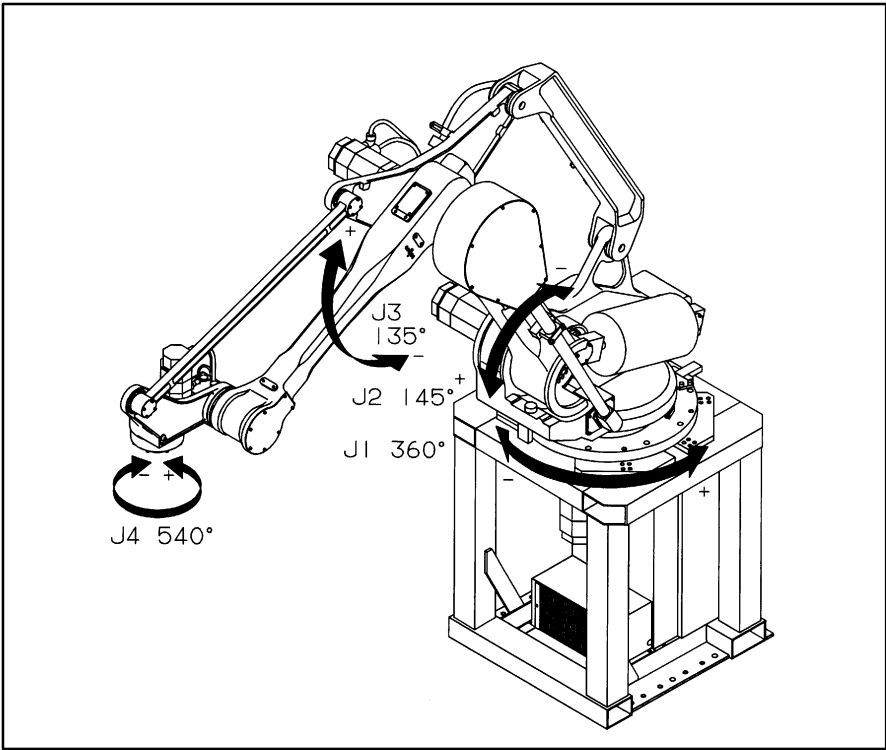


Figure 1-8. M-500 Robot Model

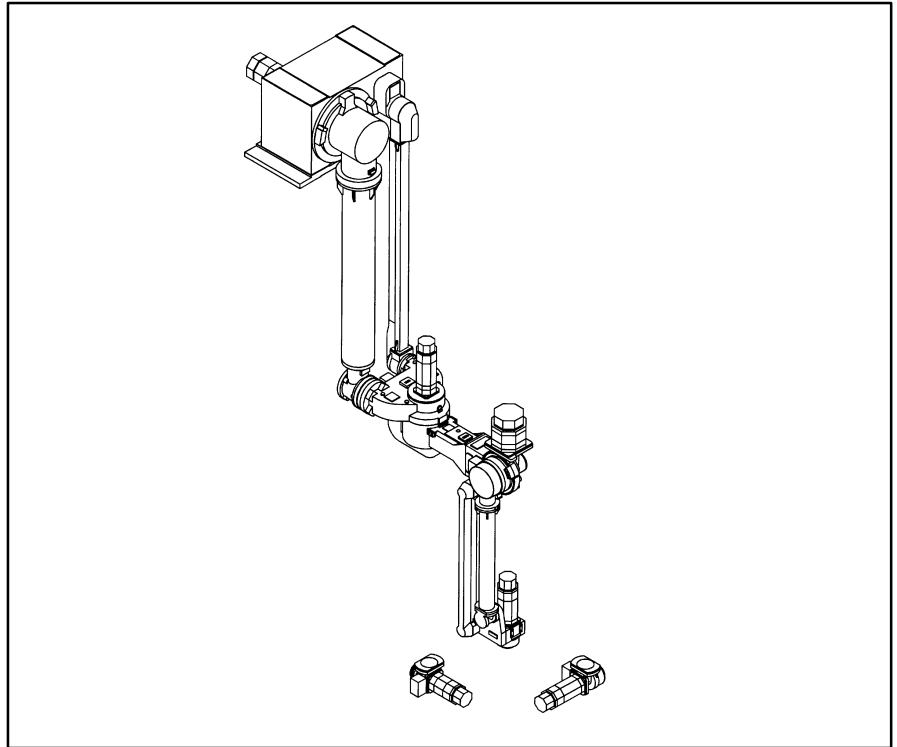


Figure 1-9. M-710i Robot Model

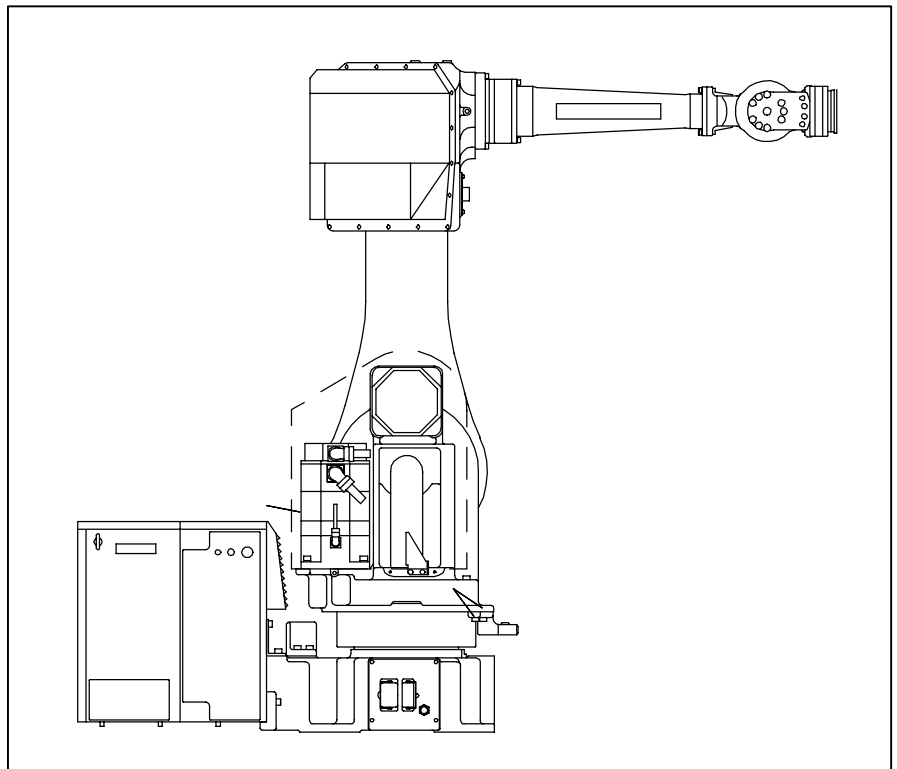


Figure 1-10. S-6 Robot Model

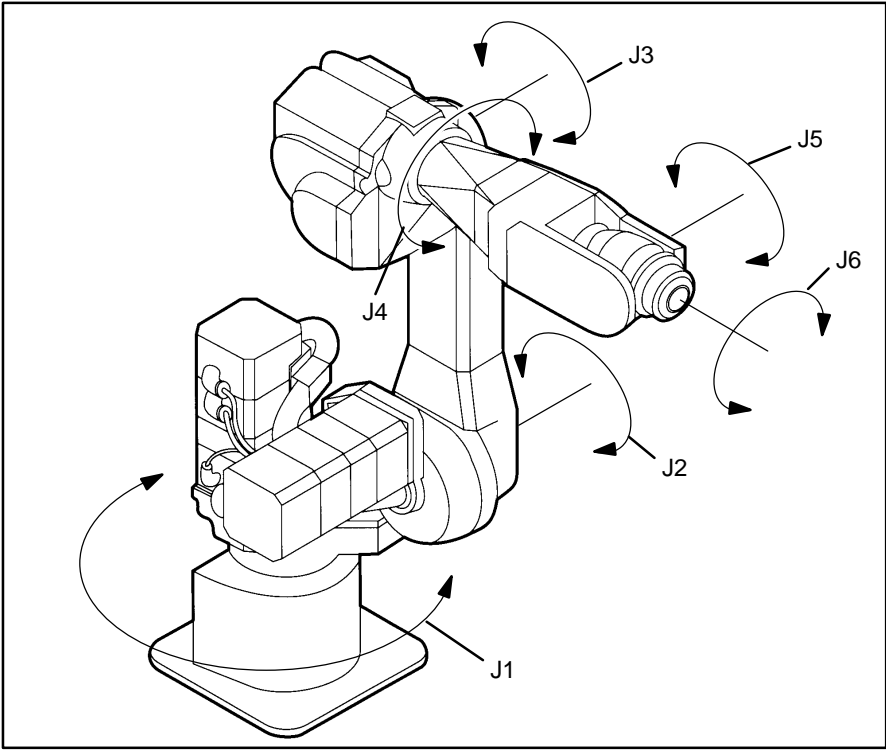


Figure 1-11. S-12 Robot Model

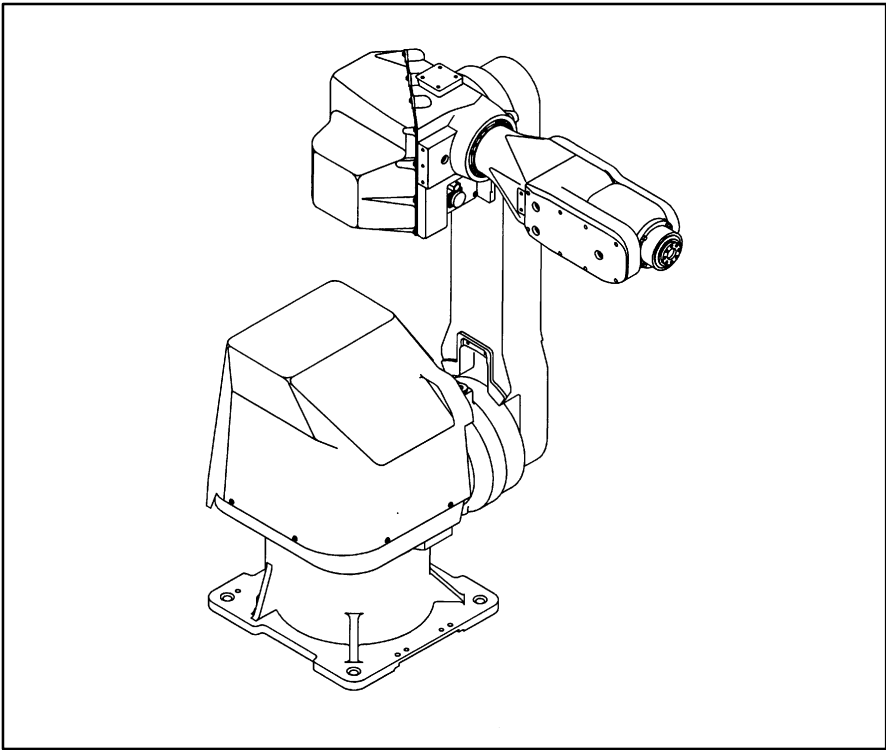


Figure 1-12. S-420iF Robot Model

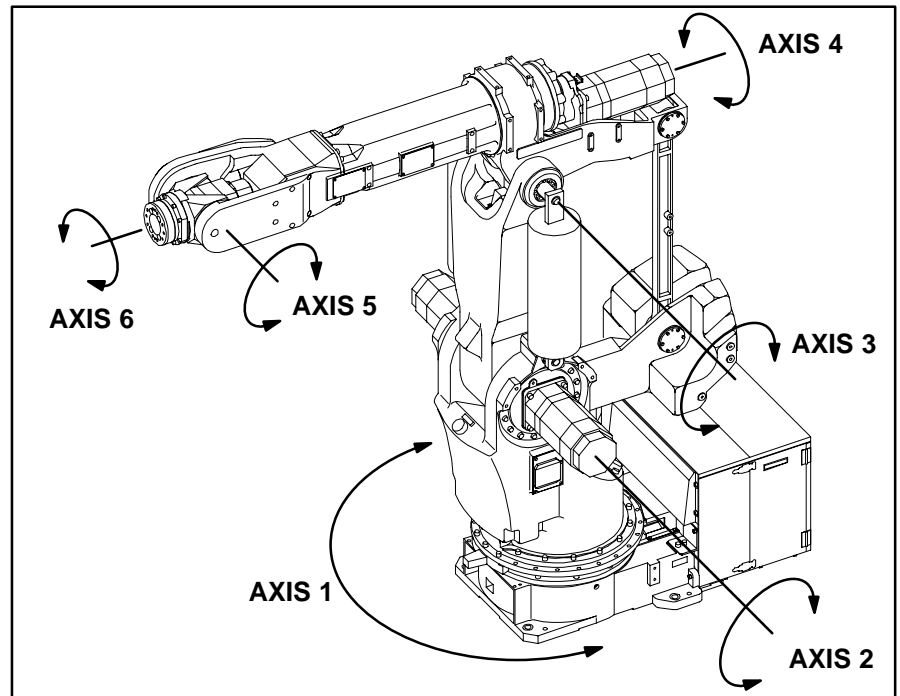


Figure 1-13. S-420iW Robot Model

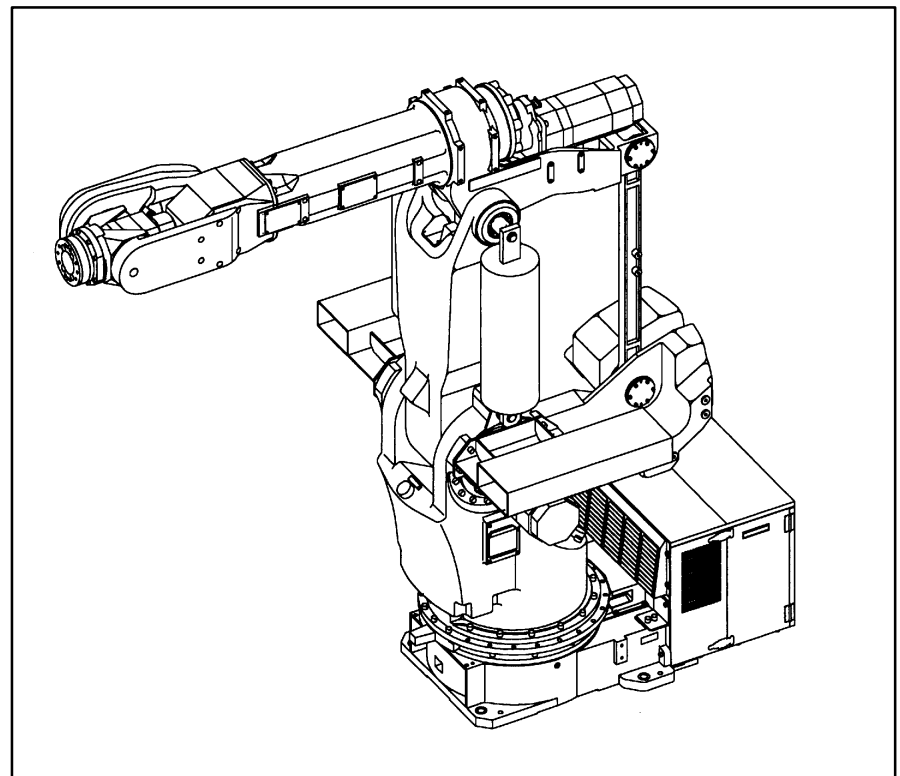


Figure 1-14. S-450S Robot Model

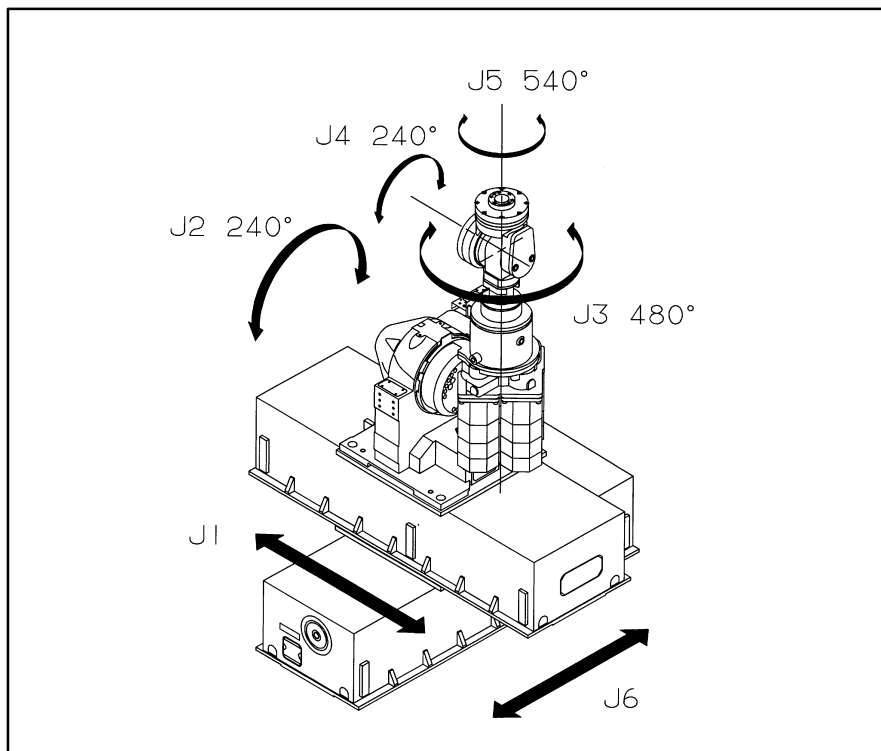


Figure 1-15. S-500 Robot Model

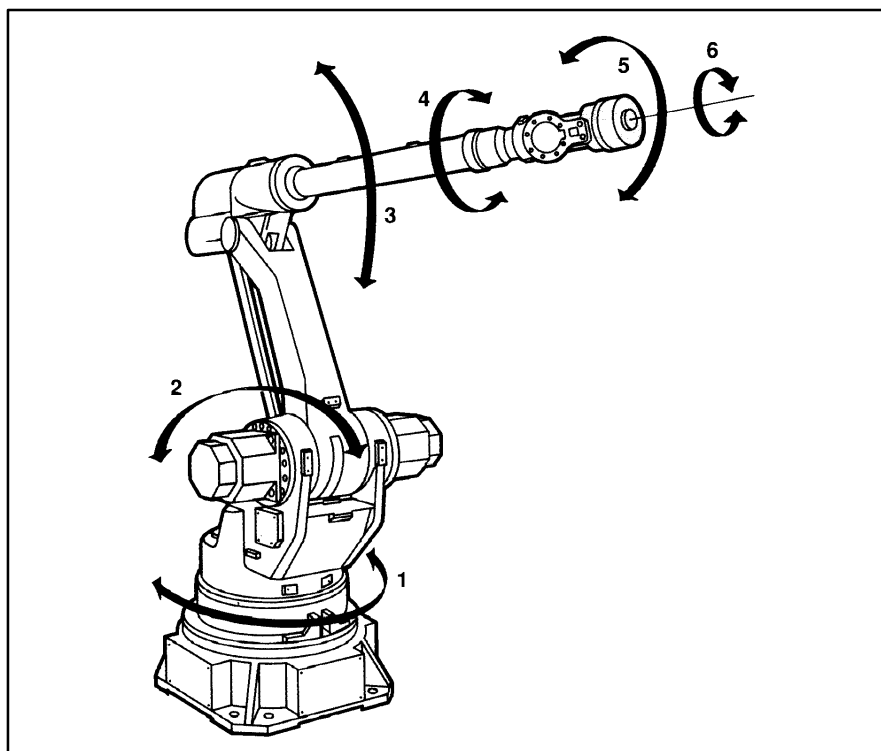


Figure 1-16. S-700 Robot Model

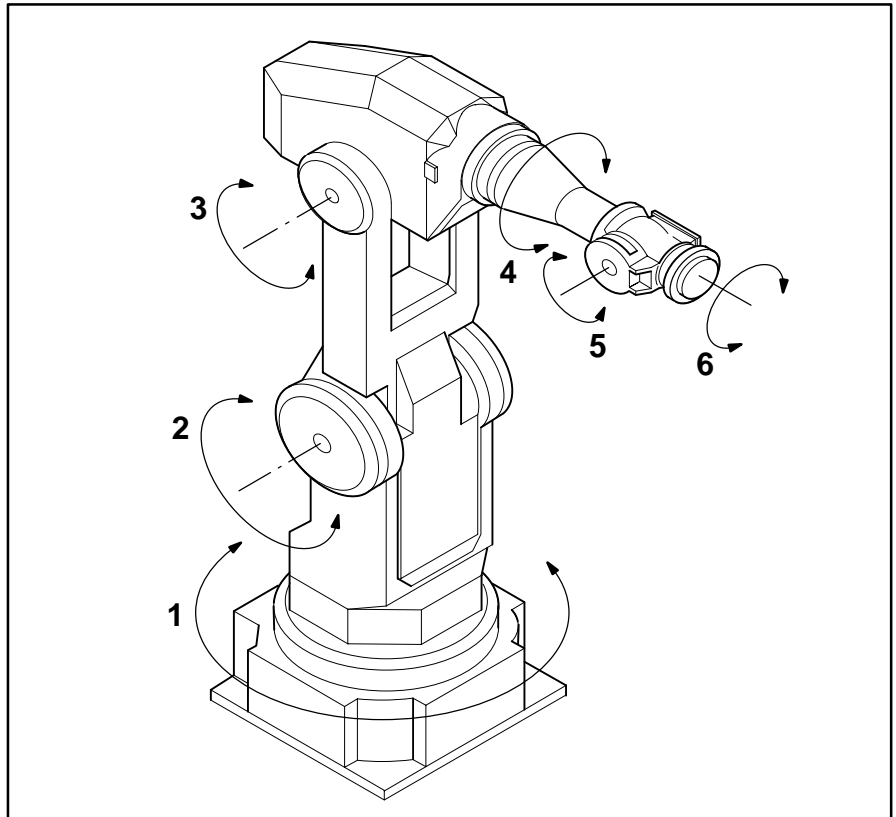


Figure 1-17. S-800 Robot Model

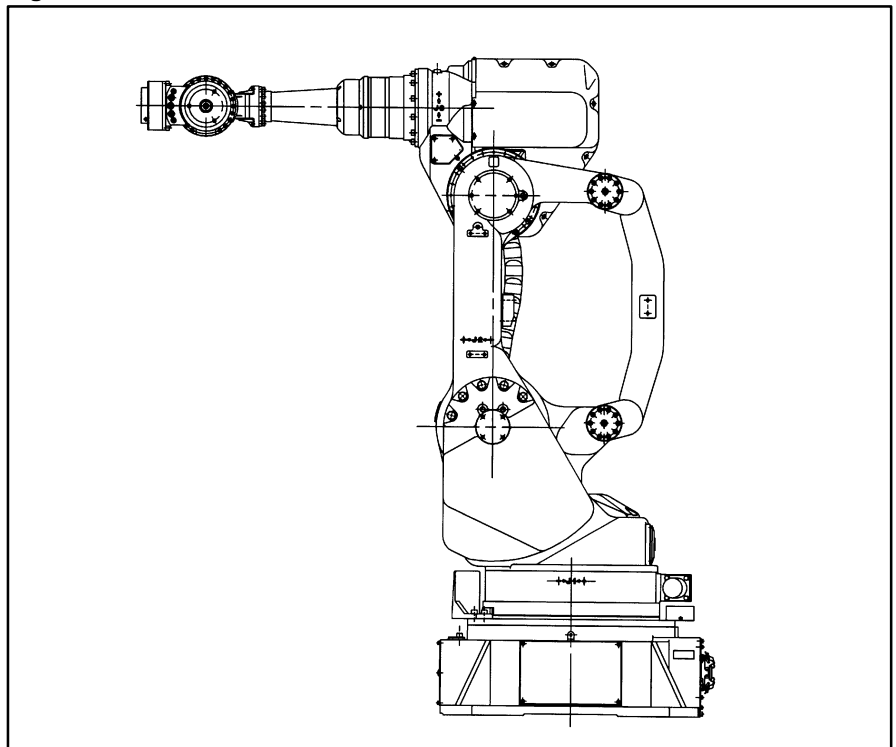
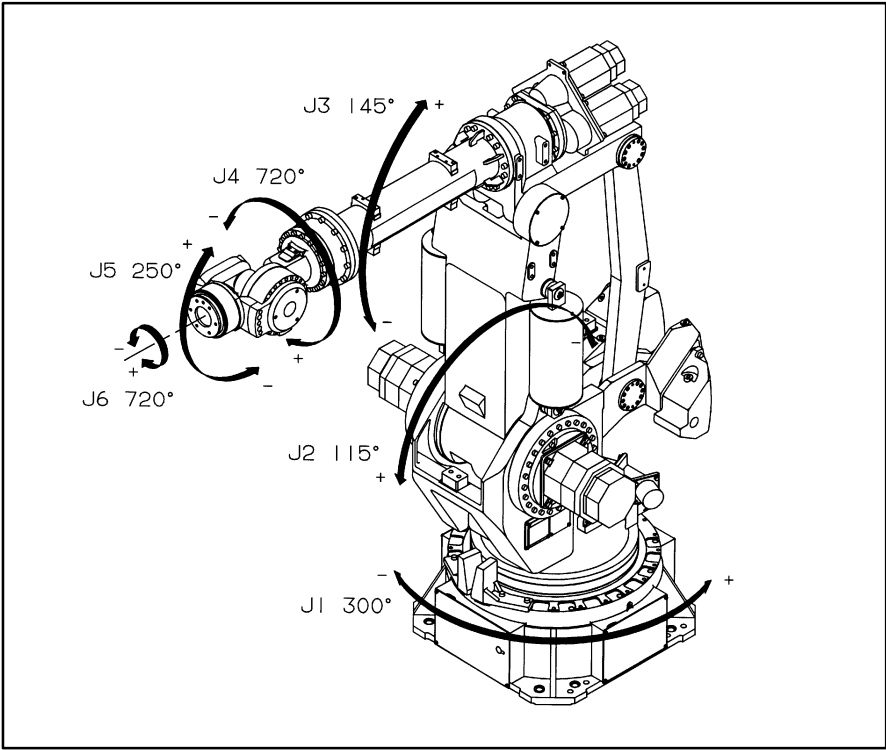


Figure 1-18. S-900 Robot Model



1.1.2 End-of-Arm Tooling

Single and double case grippers are available as options when you purchase HandlingTool. Figure 1-19 and Figure 1-20 show examples of the kinds of grippers you might use with HandlingTool.

Figure 1-19. Finger Type Gripper

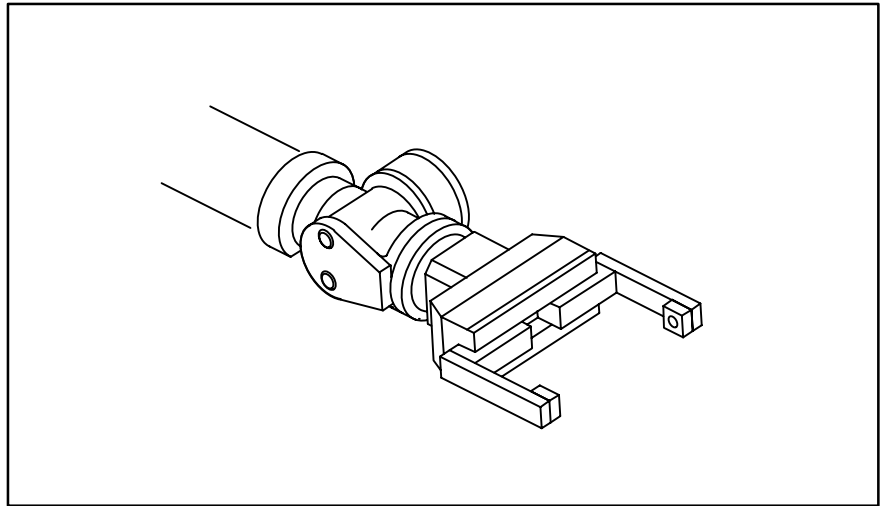
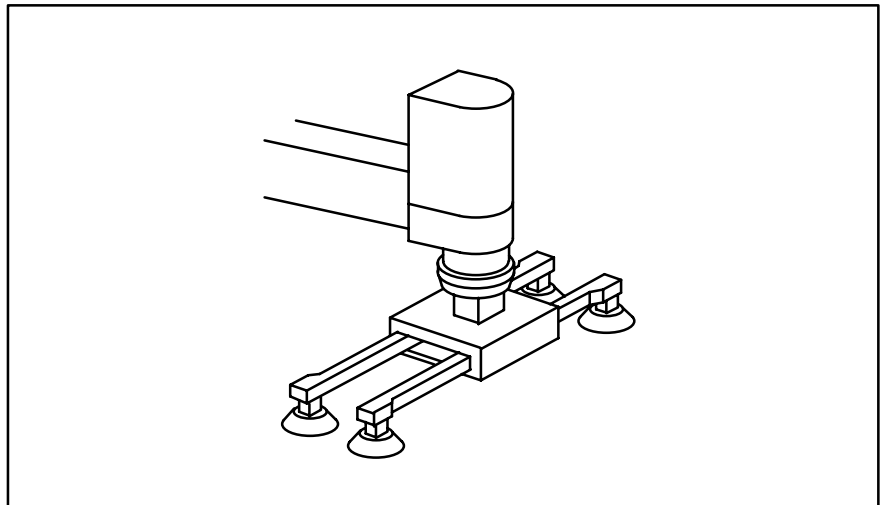


Figure 1-20. Vacuum Type Gripper



1.1.3 Extended Axes

Extended axes are the available axes controlled by the controller beyond the standard number of axes. There is a limit of three extended axes per motion group. The controller can control a maximum of 16 axes.

HandlingTool applications generally use extended axes on jobs that require a sliding axis, rail tracking, or line tracking.

Refer to Section 1.2.12 for more information about extended axes.

1.2 CONTROLLER

The R-J2 controller is attached to the robot and contains the power supply, operator controls, control circuitry, and memory that direct the operation and motion of the robot and communication with external devices. You control the robot using a teach pendant or an operator panel. Some systems contain an optional cathode ray tube/keyboard (CRT/KB) or an optional user operator panel (UOP) that provides a remote user interface to the controller.

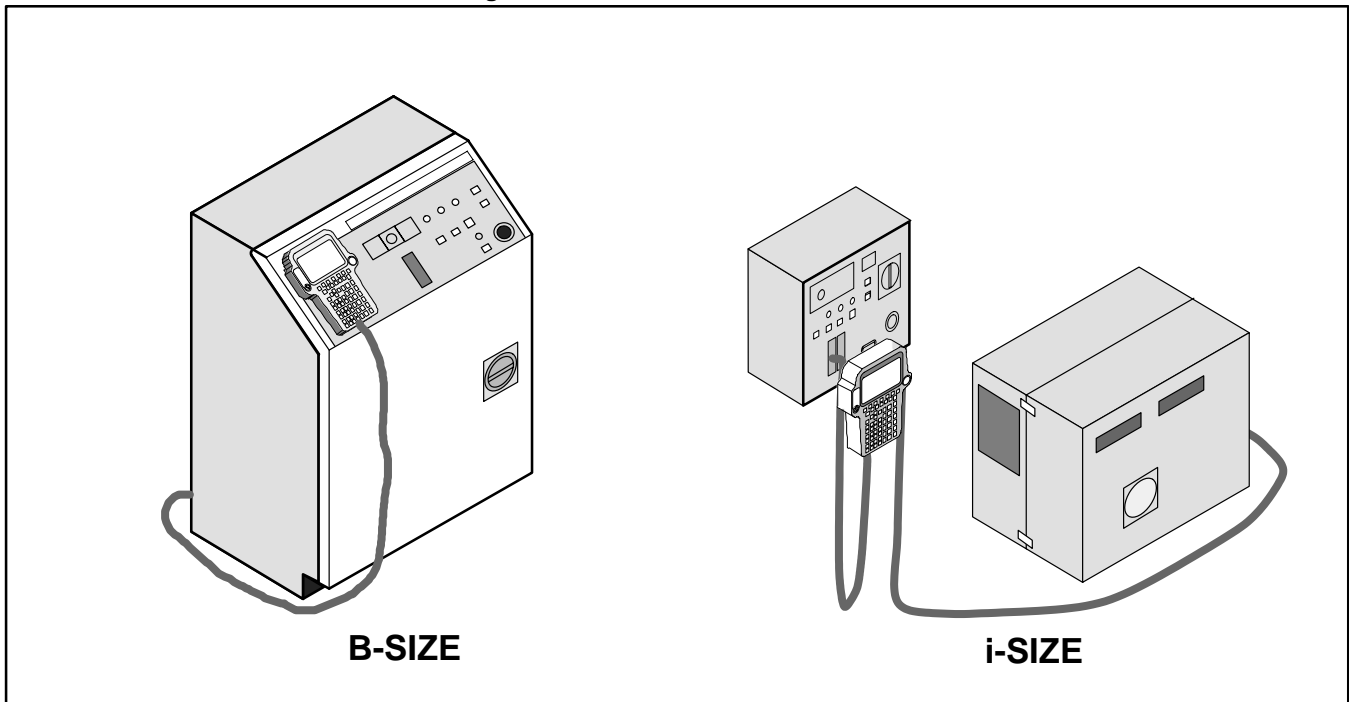
The controller has the capability of communicating with a variety of devices. Its I/O system provides an interface between the system software through I/O signals and serial communication ports to external devices. Remote I/O interfaces allow the controller to send signals to a remote device over a single cable. Consult your FANUC / FANUC Robotics representative for more information.

The motion system directs robot motion for all robot axes, including any extended axes and up to two additional motion groups.

Controller memory stores the HandlingTool software in addition to any user-defined programs and data.

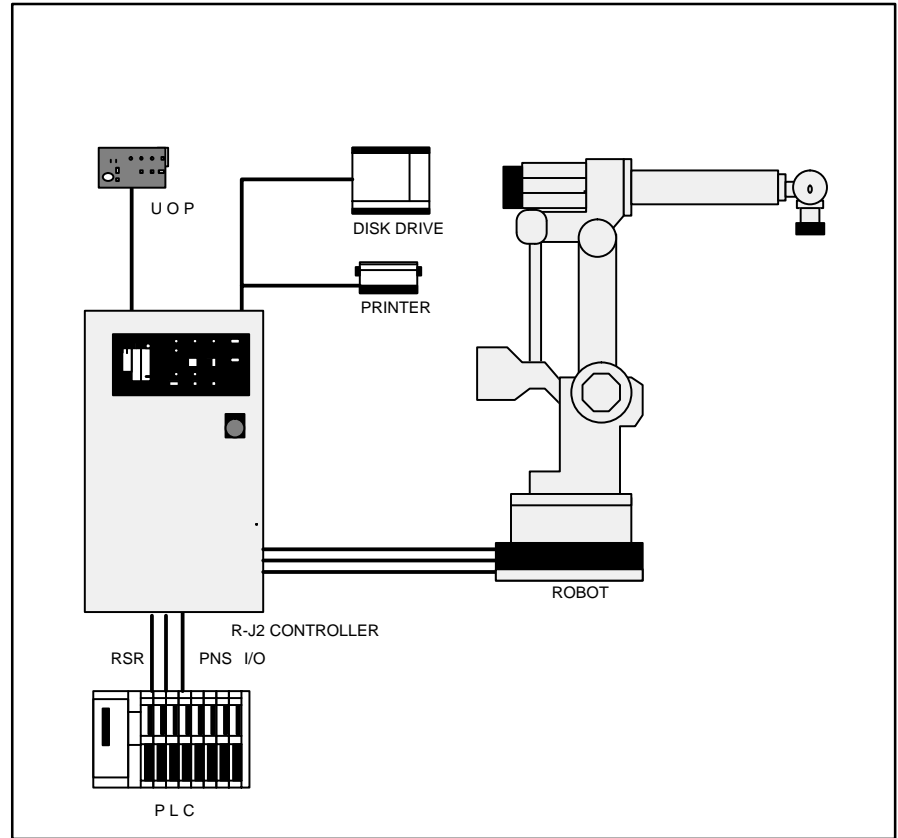
There are two sizes of controllers, as shown in Figure 1-21.

Figure 1-21. R-J2 Controllers



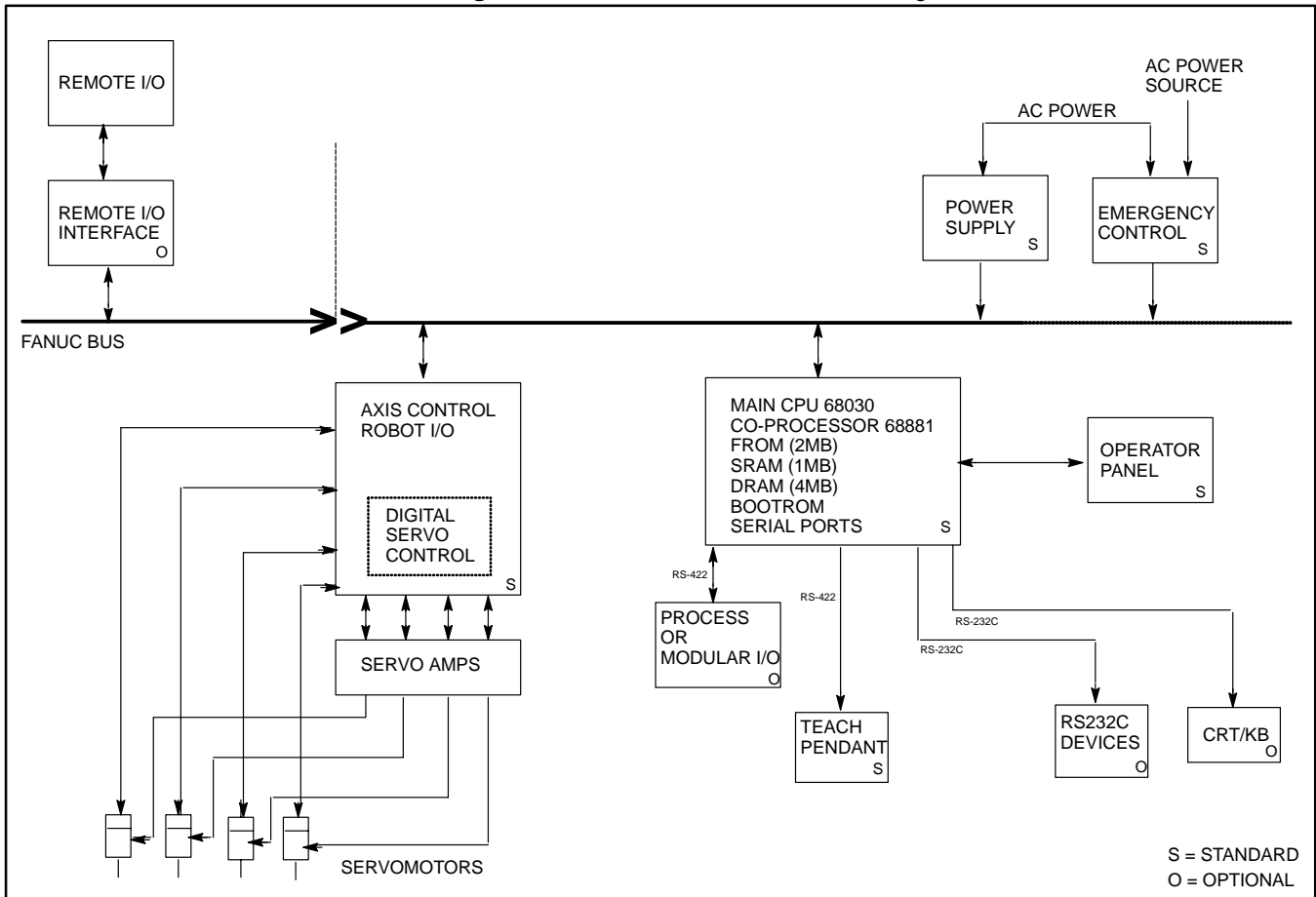
The controller provides the capability to interact with external devices. See Figure 1-22.

Figure 1-22. R-J2 Controller Capabilities



The controller is configurable internally depending on the number and types of external devices that you have in your system. See Figure 1-23.

Figure 1-23. R-J2 Controller Possible Configuration



1.2.1 Teach Pendant

The teach pendant is an operator interface device that displays the HandlingTool software menus. It is connected to the controller via a cable that plugs into either the MAIN CPU board inside the controller or, if it is a disconnectable teach pendant, to the standard operator panel.

The teach pendant is the device you use to

- Move the robot
- Create teach pendant (TP) programs
- Test programs
- Run production
- Check status

The teach pendant provides

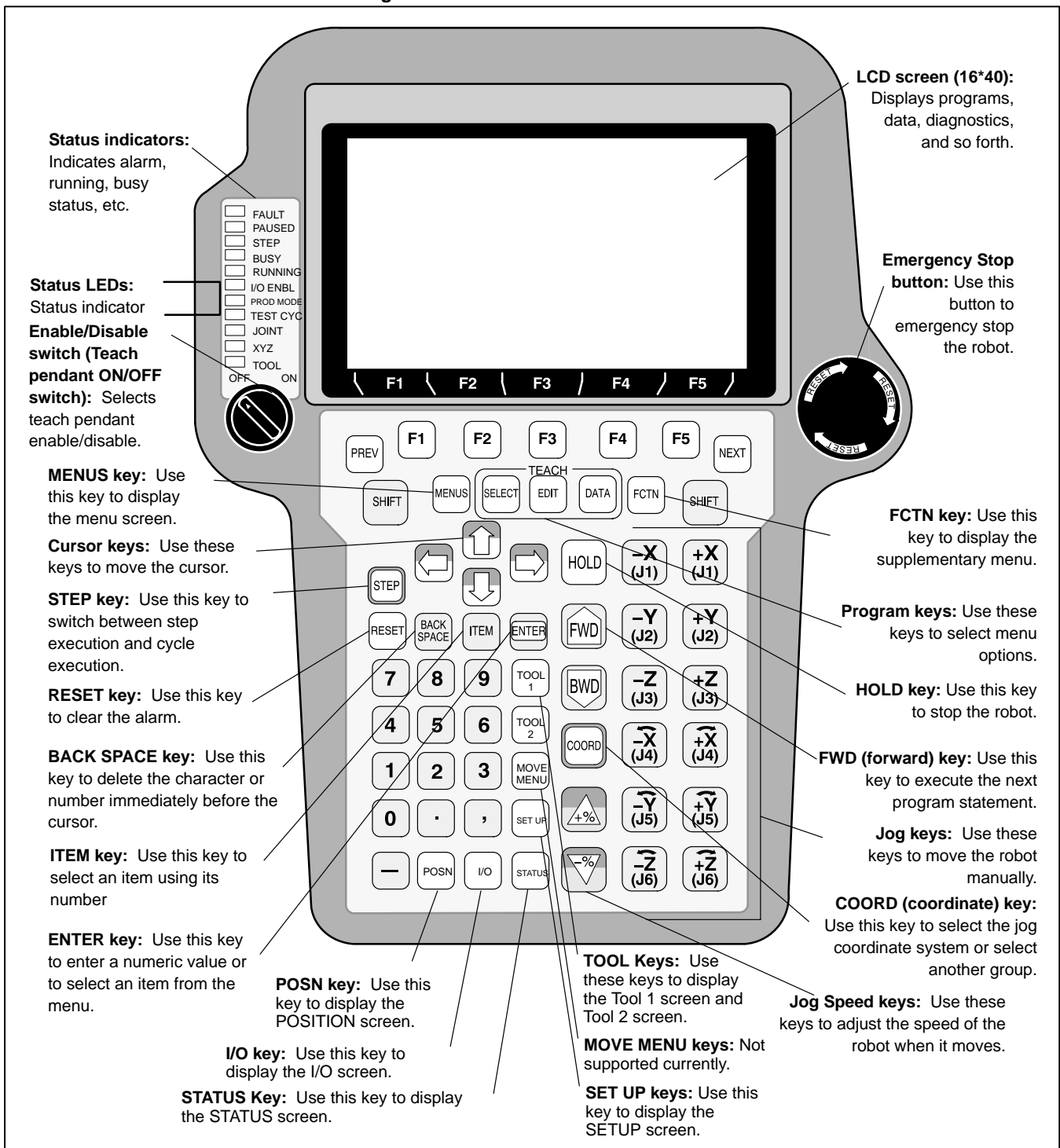
- A 16 line x 40 character teach pendant screen
- Eleven status indicators
- Teach pendant keys designed to make HandlingTool easy to use.

Figure 1-24 shows the standard teach pendant

 **CAUTION**

Do not operate the teach pendant with gloves on your hands.

Figure 1-24. Teach Pendant



Teach Pendant Keys

The teach pendant includes keys that give you control of HandlingTool. There are keys used to display software menus, select options from the teach pendant menus, help you program, move the robot, and perform specific palletizing functions.

Teach Pendant Screen

The teach pendant screen displays the HandlingTool software menus. All functions can be performed by making selections from the HandlingTool full menus.

You can alternate between display of the quick and full menus using the QUICK/FULL menu selection on the FCTN menu. The FCTN menu is displayed by pressing the FCTN key. When QUICK menus are active, the available editing functions are limited.

The full menus are a complete list of all HandlingTool menus. The QUICK menus are a partial list of specific menus.

The full menus are shown in Figure 1-25. The QUICK menus are shown in Figure 1-26.

Figure 1-25. Full Menu (pages 1 and 2)

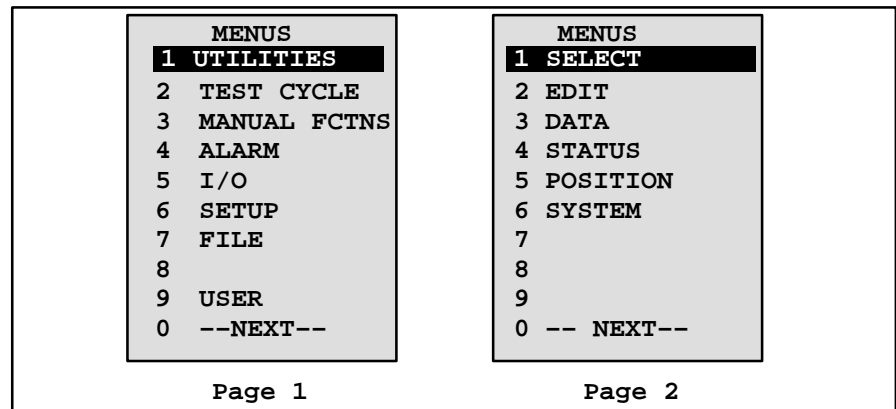
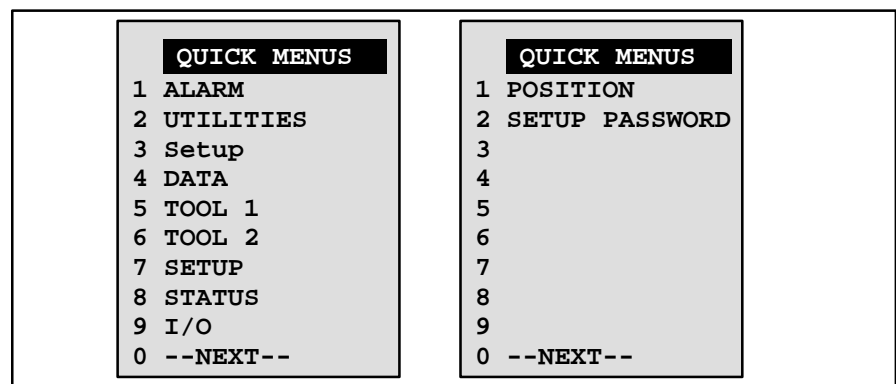


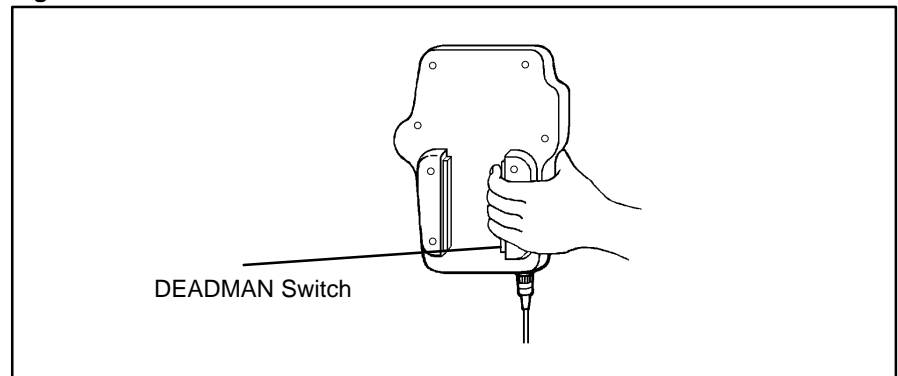
Figure 1-26. Quick Menu (pages 1 and 2)



DEADMAN Switch

The DEADMAN switch is used as an enabling device. When the teach pendant is enabled, this switch allows robot motion only while the DEADMAN switch is gripped. If you release this switch, the robot stops immediately. See Figure 1-27.

Figure 1-27. DEADMAN Switch



NOTE If you have the Control Reliable (RS-1/RS-4) option, if the DEADMAN switch is fully compressed, robot motion will not be allowed and an error occurs. This is the same as when the DEADMAN switch is released. To clear the error, press the DEADMAN switch in the center position and press RESET.

1.2.2 Standard Operator Panel (SOP)

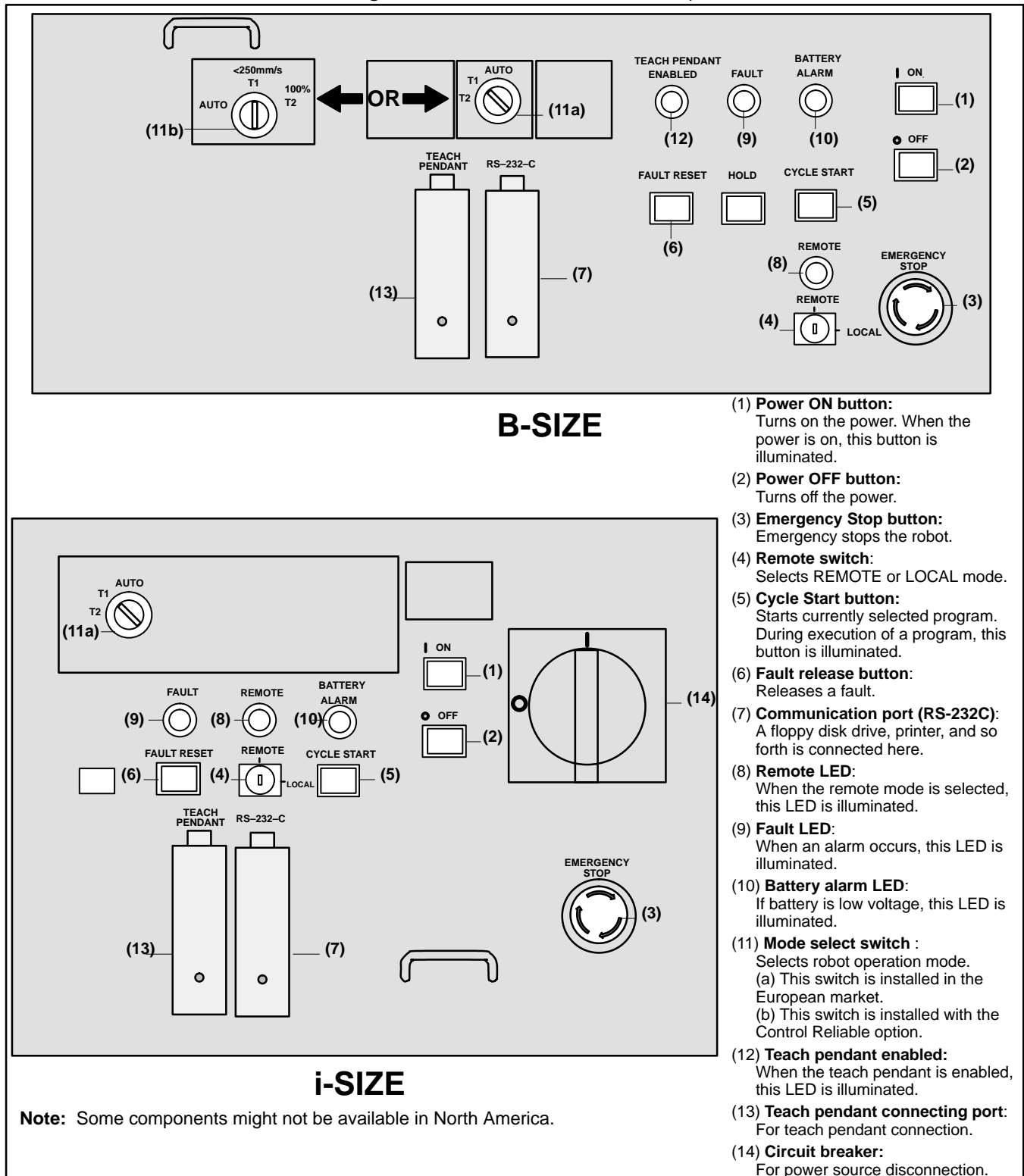
The standard operator panel (SOP) consists of buttons, keyswitches, and connector ports and is located on the front of the R-J2 controller cabinet.

The B-size controller operator panel is placed horizontally on the B-size controller. The i-size operator panel is placed independently i-size controller. Figure 1-28 displays both B- and i-size operator panels.

CAUTION

Do not operate the operator panel with gloves on your hands.

Figure 1-28. R-J2 Controller Standard Operator Panels

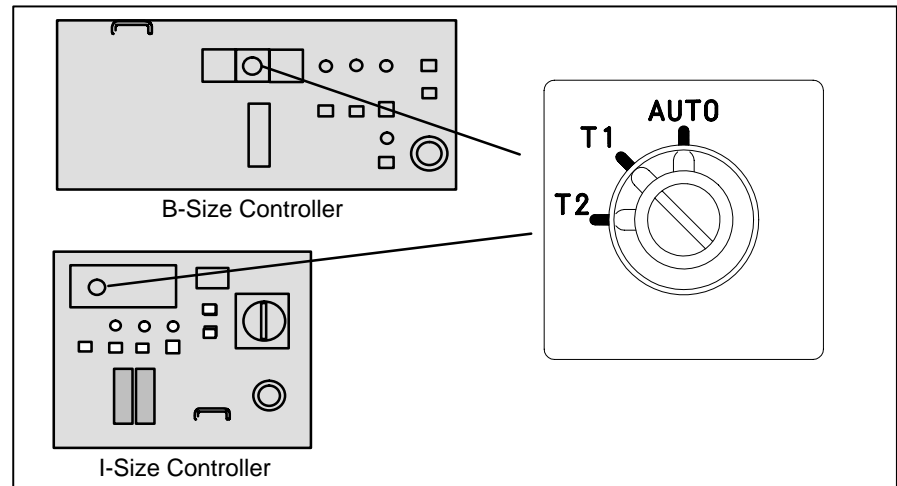


- (1) **Power ON button:**
Turns on the power. When the power is on, this button is illuminated.
- (2) **Power OFF button:**
Turns off the power.
- (3) **Emergency Stop button:**
Emergency stops the robot.
- (4) **Remote switch:**
Selects REMOTE or LOCAL mode.
- (5) **Cycle Start button:**
Starts currently selected program. During execution of a program, this button is illuminated.
- (6) **Fault release button:**
Releases a fault.
- (7) **Communication port (RS-232C):**
A floppy disk drive, printer, and so forth is connected here.
- (8) **Remote LED:**
When the remote mode is selected, this LED is illuminated.
- (9) **Fault LED:**
When an alarm occurs, this LED is illuminated.
- (10) **Battery alarm LED:**
If battery is low voltage, this LED is illuminated.
- (11) **Mode select switch :**
Selects robot operation mode.
(a) This switch is installed in the European market.
(b) This switch is installed with the Control Reliable option.
- (12) **Teach pendant enabled:**
When the teach pendant is enabled, this LED is illuminated.
- (13) **Teach pendant connecting port:**
For teach pendant connection.
- (14) **Circuit breaker:**
For power source disconnection.

Mode Select Switch (for European Controllers)

The MODE SELECT KEY SWITCH is installed on the operator panel. This switch is required in European controllers. You can select one of the operation modes using this switch. The selected operation mode can be locked by removing its key. When the mode is changed using this switch, the robot system stops and a message is displayed on the teach pendant screen. See Figure 1-29.

Figure 1-29. Mode Select Switch (European Controller)



T1: Test Mode 1

- Programs can be activated from the teach pendant only. However, programs cannot be activated while the teach pendant is disabled.
- The robot cannot operate at speeds higher than 250 mm/s. At this time, the speed override can be increased up to 100% during jogging. When a program is executed, however, the override is limited to the minimum value set in `$$SCR_GRP[.].$JOGLIM_JONT[.]`.
- Safety equipment, including the fence, is disabled.

T2: Test Mode 2

- Programs can be activated from the teach pendant only. However, programs cannot be activated while the teach pendant is disabled.
- The robot can operate at the specified maximum speed.
- Safety equipment, including the fence, is disabled.

AUTO: Automatic Mode

- The selection of automatic mode is added to the conditions that must be satisfied to enable the activation of programs from remote devices connected through the peripheral I/O. Other specifications for activation are the same as when this option is not used.
- Safety equipment, including the fence, is enabled.

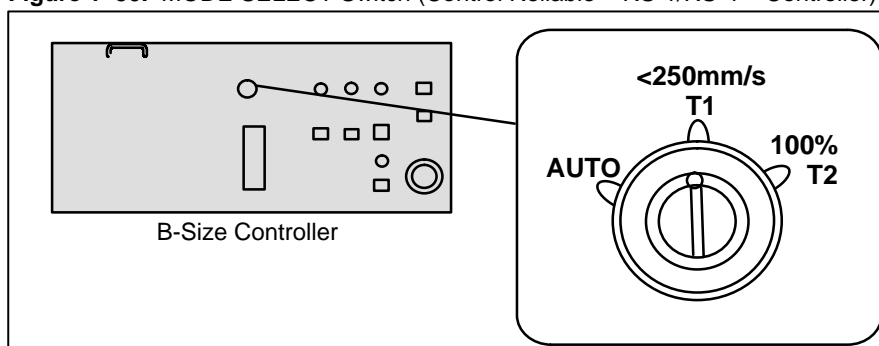
NOTE

- When the operation mode is changed using the mode select switch, a message is displayed on the teach pendant screen and the robot is paused.
- When an erroneous input (invalid combination) is detected on the signal lines for the Mode select switch, programs can be activated only from the teach pendant. The robot cannot operate at speeds of more than 250 mm/s. If a program is being executed at that time, it is forcibly terminated.

Mode Select Switch (for Controllers with the Control Reliable option)

The MODE SELECT switch is a keyswitch installed on the operator panel or operator box on controllers that have the Control Reliable (RS-1/RS-4) option. You use the MODE SELECT switch to select the most appropriate way to operate the robot, depending on the conditions and situation. The operation modes are AUTO, T1, and T2. See Figure 1-30.

Figure 1-30. MODE SELECT Switch (Control Reliable – RS-1/RS-4 – Controller)



When you change the mode using the MODE SELECT switch, a message is displayed on the teach pendant screen and the robot is paused. You can also lock the keyswitch in the AUTO or T1 modes by removing the key from the switch. You cannot remove the key from the keyswitch when the key is in the T2 position.

NOTE If you change the mode from T1 or T2 to AUTO and the DEADMAN switch is pressed, a system error will occur and the mode will not change to AUTO until the DEADMAN switch is released

The operation modes you can select using the MODE SELECT switch are described in the following sections.

T1 (<250mm/s): Test Mode 1

Program activation – Programs can be activated from the teach pendant only. However, programs can be activated only when the teach pendant is enabled and when the DEADMAN switch is in the center position.

Robot speed

- During Cartesian jogging, Cartesian speed is less than 250 mm/sec and joint speed is less than 10% of the maximum joint speed.
- During joint jogging, face plate speed is less than 250 mm/sec.
- During program test run, the override is limited to 5%.

Safety equipment – The safety fence is bypassed.

Locking the mode – You can lock the switch in T1 mode by removing the key from the switch.

Possible errors

- If you turn the teach pendant ON/OFF switch to OFF when the switch is in T1 mode, the robot stops and an error message is displayed. To remove the error, turn the teach pendant ON/OFF switch to ON and press RESET.
- If you have set the singularity stop system variable, \$PARAM_GROUP[n].\$T1T2_SNGSTP, to TRUE, the robot will stop at singularity points while in T1 mode. If you change the value of this variable, you must cycle power for the change to take effect.

T2 (100%): Test Mode 2

Program activation – Programs can be activated from the teach pendant only. However, programs can be activated only when the teach pendant is enabled and the DEADMAN switch is in the center position.

Robot speed

- During Cartesian jogging, Cartesian speed is less than 250 mm/sec and joint speed is less than 10% of the maximum joint speed.
- During joint jogging, face plate speed is less than 250 mm/sec.
- During program test run, full program speed is allowed, and the override can be changed from low to 100%.

Safety equipment – The safety fence is bypassed.

Locking the mode – You cannot lock the switch in T2. You cannot remove the key from the switch in this mode.

Possible errors

- If you turn the teach pendant ON/OFF switch to OFF when the switch is in T2 mode, the robot stops and an error message is displayed. To remove the error, turn the teach pendant ON/OFF switch to ON and press RESET.
- If you have set the singularity stop system variable, \$PARAM_GROUP[n].\$T1T2_SNGSTP, to TRUE, the robot will stop at singularity points in while T2 mode. If you change the value of this variable, you must cycle power for the change to take effect.

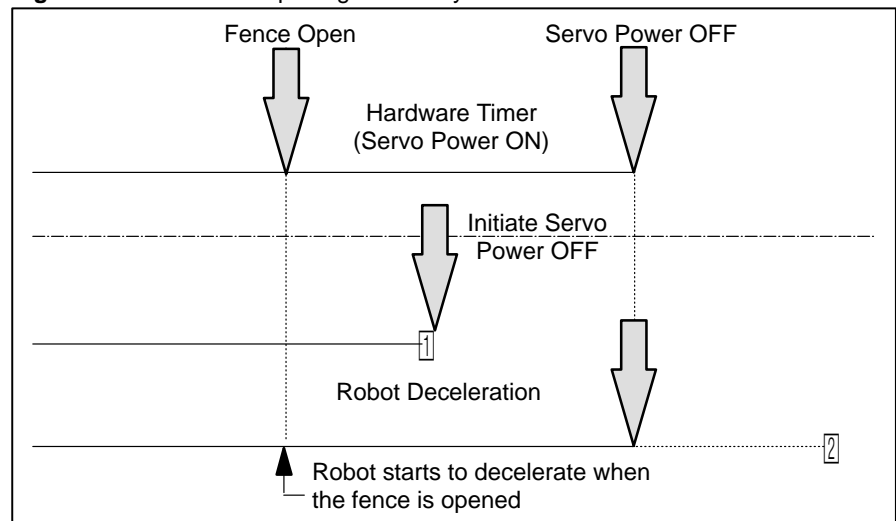
AUTO: Automatic Mode

Program activation – You must select AUTO mode and satisfy all other required conditions to enable the activation of programs from remote devices connected through the peripheral I/O. Other required conditions are the same as when the Control Reliable (RS-1/RS-4) option is not used. When the switch is in AUTO mode, you cannot start programs using the teach pendant.

Robot speed – The robot can be operated at the specified maximum speed.

Safety equipment – The safety fence is monitored. If the safety fence is opened during program execution (Figure 1-31):

- **Case 1** – If the robot deceleration time is less than the *hardware timer*, then the robot will decelerate to a stop. When the robot stops, servo power OFF is initiated.
- **Case 2** – If the robot deceleration time is greater than the hardware timer, then the robot will decelerate for the duration of the hardware timer and then stop abruptly when the hardware timer expires. When the hardware timer expires, servo power is turned OFF.

Figure 1-31. Effect of Opening the Safety Fence While in AUTO Mode

The system variable `$PARAM_GROUP.$LC_QSTP_ENB` defines whether the condition specified by the condition monitor (condition handler) function will be triggered during robot deceleration. By default, the condition, if it exists, is triggered during deceleration (`$LC_QSTP_ENB = TRUE`). When `$LC_QSTP_ENB = FALSE`, a condition, if it exists, is not triggered during deceleration.

Refer to the *FANUC Robotics SYSTEM R-J2 Controller Software Reference Manual* for more information on these system variables.

Locking the mode – You can lock the switch in AUTO mode by removing the key from the switch.

Possible errors

- If you turn the teach pendant ON/OFF switch to ON when in AUTO mode, the robot stops and an error message is displayed. To remove the error, turn the teach pendant ON/OFF switch OFF and press RESET.
- If you press the DEADMAN switch when in AUTO mode, nothing happens.
- If you have set the singularity stop system variable, `$PARAM_GROUP[n].$AUTO_SNGSTP`, to FALSE, the robot will pass through singularity points while in AUTO mode. If you change the value of this variable, you must cycle power for the change to take effect.

1.2.3 User Operator Panel (UOP)

Your system might be equipped with a user operator panel (UOP). A UOP is a customized operator panel that is wired to the controller. It can be a custom control panel, a cell controller, or a host computer. Your company should provide the information necessary for operating this panel.

1.2.4 CRT/KB

The CRT/KB provides an optional alternative operator device to the teach pendant. The external CRT/KB connects to the controller via a cable.

The CRT/KB can be used as an alternative menu to display the HandlingTool software. The CRT/KB allows you to perform most teach pendant functions except those that are related to robot motion. Functions that cause robot motion can only be performed using the teach pendant.

NOTE The built-in CRT/KB is not available for the B-size or i-size controllers.

1.2.5 Emergency Stop Devices

This robot has the following emergency stop devices.

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

When the EMERGENCY STOP button is pushed, the robot stops immediately in all cases. The external emergency stop outputs or inputs the emergency stop signal for peripheral devices (such as a safety fence or gate). The signal terminal is on the controller and inside the operator box.

1.2.6 Robot Stop Variation (for European Controllers)

When the emergency stop switch on operator panel, operator box, or teach pendant is pressed, the robot stops immediately.

In other cases (excluding when emergency stop switch is pressed), there are following situations when an emergency stop condition is created by the combination of operation mode selection, teach pendant enable and disable, DEADMAN SWITCH, and safety fence open and close. Refer to Table 1-1.

Table 1-1. Robot Stop Variation for European Controllers

Mode	TP-Enable	DEADMAN	Fence	Status
Auto	enable	grip	open	EMG-stop
			close	available
		release	open	EMG-stop
			close	EMG-stop
	disable	grip	open	EMG-stop
			close	available
		release	open	EMG-stop
			close	available
T1/T2	enable	grip	open	available
			close	available
		release	open	EMG-stop
			close	EMG-stop
	disable	grip	open	n.a
			close	n.a
		release	open	EMG-stop
			close	EMG-stop

EMG-stop : Emergency stop

available : Robot is available

n.a : Robot is not available

1.2.7 Robot Stop Variation (for Control Reliable (RS-1/RS-4) option only)

When the EMERGENCY STOP button on operator panel, operator box, or teach pendant is pressed, the robot stops immediately.

An emergency stop condition can be created not only when the EMERGENCY STOP button is pressed, but also by a combination of operation mode selection, teach pendant ON/OFF switch, DEADMAN switch, and safety fence open and close. Refer to Table 1-2.

NOTE If you have the Control Reliable (RS-1/RS-4) option, and the DEADMAN switch is fully compressed, robot motion will not be allowed. This is the same as when the DEADMAN switch is released.

Table 1-2. Robot Servo Status for Control Reliable (RS-1/RS-4) Option

Mode	TP-ON/OFF Switch	DEADMAN Switch	Fence	SERVO Status	Motion Possible
AUTO	ON	pressed (*)	open	OFF	No
			close	ON	No
		released (*) or pressed extremely firmly	open	OFF	No
			close	ON	No
	OFF	pressed (*)	open	OFF	No
			close	ON	Yes
		released (*) or pressed extremely firmly	open	OFF	No
			close	ON	Yes
T1 or T2	ON	pressed	open	ON	Yes
			close	ON	Yes
		released or pressed extremely firmly	open	OFF	No
			close	OFF	No
	OFF	pressed	open	OFF	No
			close	OFF	No
		released or pressed extremely firmly	open	OFF	No
			close	OFF	No

* The DEADMAN switch on the teach pendant is ignored in AUTO mode.

1.2.8 Communications

The controller has the capability of serial communication using:

- RS-422 serial ports, which can be used for
 - Teach pendant
 - Remote teach pendant
 - Custom serial interdevice link
- RS-232-C serial ports, which can be used for
 - CRT/KB
 - FANUC Robotics Industrialized Terminal
 - DEC VT-220 terminal
 - IBM PC compatibles
 - PS-100, PS-110, or PS-200 disk drives
 - FANUC FLOPPY CASSETTE DISK
 - FANUC Handy File
 - Printers
 - Debug monitor
- GEFANUC Genius I/O Network Interface
- Allen-Bradley Remote I/O Interface
- DeviceNet Interface
- Ethernet for Local Area Networks (LANs): protocols available are FTP, Ethernet Controller Backup and Restore, and MOTET

R-J2 controller (B-Size) available port configurations include:

- **Standard** port configuration, which consists of
 - One RS-422 serial port
 - One RS-232-C serial port
- **Optional** port configurations, which consist of
 - Three RS-232-C and one RS-422 serial ports

1.2.9 Input/Output (I/O)

The I/O system provides the interface between the controller, teach pendant, robot, and any other external device in your workcell. Controller I/O can consist of the following kinds of I/O:

- User Operator Panel (UOP) Inputs (UI)
- User Operator Panel (UOP) Outputs (UO)
- Standard Operator Panel (SOP) Inputs (SI) (Option)
- Standard Operator Panel (SOP) Outputs (SO) (Option)
- Robot Inputs (RI) (Option)
- Robot Outputs (RO) (Option)
- Digital Inputs (DI)
- Digital Outputs (DO)
- Group Inputs (GI)
- Group Outputs (GO)
- Analog Inputs (AI) (Option)
- Analog Outputs (AO) (Option)
- PLC Inputs
- PLC Outputs

These kinds of I/O are provided by devices, including

- Process I/O (Option)
- Modular I/O (Model A and Model B)
- GEFanuc Genius I/O Network
- Allen Bradley Remote I/O
- DeviceNet
- Ethernet

The **quantity** of I/O can change, except for UOP and SOP I/O signals, which are fixed.

1.2.10 Remote I/O Interfaces

The controller has the capability to use certain signals from a remote device. These signals can include

- UOP signals
- Safety fence
- RSR and PNS
- External Emergency stop

1.2.11 Motion

The R-J2 robot system uses the motion system to control motion. The motion system regulates the characteristics of the movement including path trajectory, acceleration and deceleration, termination and robot speed.

In robotic applications, single segment motion is the movement of the tool center point (TCP) from an initial position to a desired destination position.

Motion Type

There are three different types of motion: linear, circular, and joint. You use these motion types to perform certain tasks. For example, you use linear motion if the robot must move in a straight line between two positions. You use circular motion when the positions must be along the arc of a circle. Joint motion is generally the motion type used at each position when it is not important how the robot moves from position to position.

Termination Type and Speed

Termination type can be specified as fine or continuous. Speed can be specified in either degrees of angle units or length for a given or total time to execute the move.

Motion Groups

The R-J2 controller optionally allows you to create up to three motion groups. By default, one motion group is always available. Additional motion groups can be set up to perform tasks that are executed simultaneously with those of the robot.

1.2.12 Extended Axes

Extended axes are the available axes controlled by the controller beyond the standard number of robot axes. There is a maximum of three extended axes in any motion group. The controller can control a maximum of 16 axes.

1.2.13**Controller Backplane**

The following kinds of backplanes are available with the R-J2 controller:

- 3-slot backplane – i-size and B-size controllers
- 5-slot backplane – B-size controller only

3-Slot Backplane

The 3-slot backplane comes equipped with the

- Power supply
- Main CPU
- Process I/O (option)

5-Slot Backplane

The 5-slot backplane has the same components as the 3-slot backplane, plus two spare slots. The two remote I/O optional open slots can be used to customize the controller for your application. The 5-slot backplane is available only on the B-size controller.

1.2.14**Memory**

The following kinds of internal controller memory are available:

- D-RAM (Dynamic Random Access Memory)
- CMOS RAM (Random Access Memory)
- Flash ROM (FROM) (Read Only Memory)

In addition, the controller is capable of storing information externally.

D-RAM

DRAM is volatile RAM that is used for:

- Working memory for the system
- Loaded KAREL programs
- Most KAREL variables

CMOS RAM

CMOS RAM is battery-backed RAM that is used for:

- HandlingTool software
- User programs
- Variable data

Flash ROM

Flash ROM, or FROM, is non-volatile memory that contains HandlingTool and does not change.

External Storage

The ability to back up and store files on external memory such as floppy disks, and on FANUC's Off-line personal computer (OLPC) also exists.

Refer to Chapter 9 for more information.

1.3 HANDLINGTOOL SOFTWARE

HandlingTool software works in conjunction with the robot and the R-J2 controller to allow you to:

- **Set up** information required for the application.
- **Program** your application.
- **Test** your program.
- **Run production.**
- **Display** and monitor process information.

Other tools such as program and file management capabilities help you to maintain your system before, during, and after the production stage.

The **menu maps** in Section 1.4 illustrate how to access each software function.

1.3.1 Set Up

HandlingTool software provides the components necessary to set up all the information required for your application. It also provides the necessary commands for you to set up how you want your programs to run during production.

HandlingTool software also allows you to work directly with external devices such as cell controllers. You will need to set up your equipment before you begin your application.

1.3.2 Program

An application program is a combination of instructions that, when executed in a sequence, will complete your material handling task. Refer to Chapter 6, “Program Elements,” for more information.

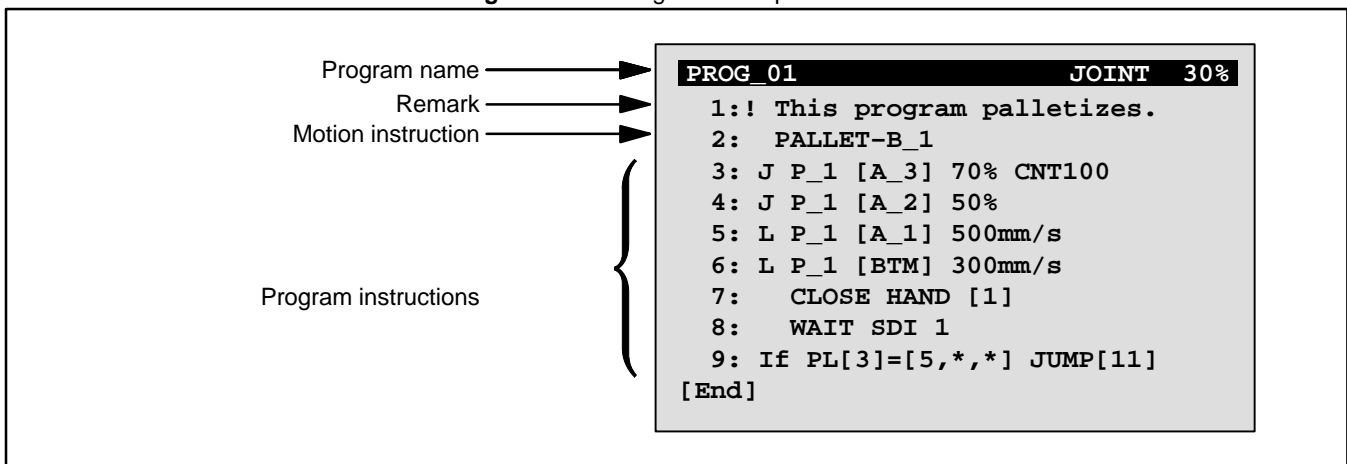
The HandlingTool software allows you to create and modify an application program to consist of

- *HandlingTool* instructions to palletize.
- Motion instructions to position the robot or workpiece in the appropriate locations in the workcell.
- Register instructions to store numerical program information.
- Position register instructions to manipulate program positional information.
- I/O instructions to send signals to and receive signals from equipment in the workcell.
- Branching and label instructions, (JUMP and IF/SELECT) to control the direction and order of program flow.
- Wait instructions to delay program execution.
- Skip instructions to move the robot until a signal is received. After the signal is received, stop and branch to the specified statement.

- Offset and tool offset instructions to compensate for variations in the workpiece.
- Multiple control instructions to control different motion groups and different programmed tasks.
- Macro command instructions to perform specific, frequently used functions.
- Program control instructions to direct program execution.
- Miscellaneous instructions to allow functions such as writing messages to the screen.
- Sensor instructions to control sensors
- Motion group instructions to use motion groups.
- Position register look-ahead instructions to control motion execution.
- Condition monitor instructions to monitor I/O, register, and alarm conditions during program execution.
- Payload instructions to set the appropriate payload schedule.
- Collision Guard instructions to use Collision Guard in a program.

Figure 1-32 displays a typical palletizing application program.

Figure 1-32. Program Example



1.3.3 Test Program

After you have set up HandlingTool and successfully created a program, you must test your application to be sure it runs correctly. Refer to Chapter 7 for more information.

Testing the program is an important step in creating a successful application program. Be certain to test the program thoroughly before running it in production.

1.3.4 Run Production

Running production is the final step in executing an application program. It consists of

- Adjusting the RSR or PNS job queue to specify which programs to run
- Adjusting program data
- Performing recovery and restart procedures
- Running the application program

Refer to Chapter 7 for more information.

1.4 MENU MAPS

This section contains menu maps that illustrate how to display each screen on the teach pendant. Figure 1-33 shows the full menus and indicates the section that contains the menu map for each menu item. Figure 1-34 shows the FCTN menu.

NOTE The menu maps will vary depending on the options installed in your system. Some items included in the menu maps will not be available unless you have purchased and installed the appropriate software option.

NOTE If you do not have multiple groups installed, you will not see the GROUP function key as displayed in the menu map sections. Also, if you do not have any extended axes installed, you will not see the PAGE function key as displayed in the menu map sections.

Figure 1-33. Full Menu (pages 1 and 2)

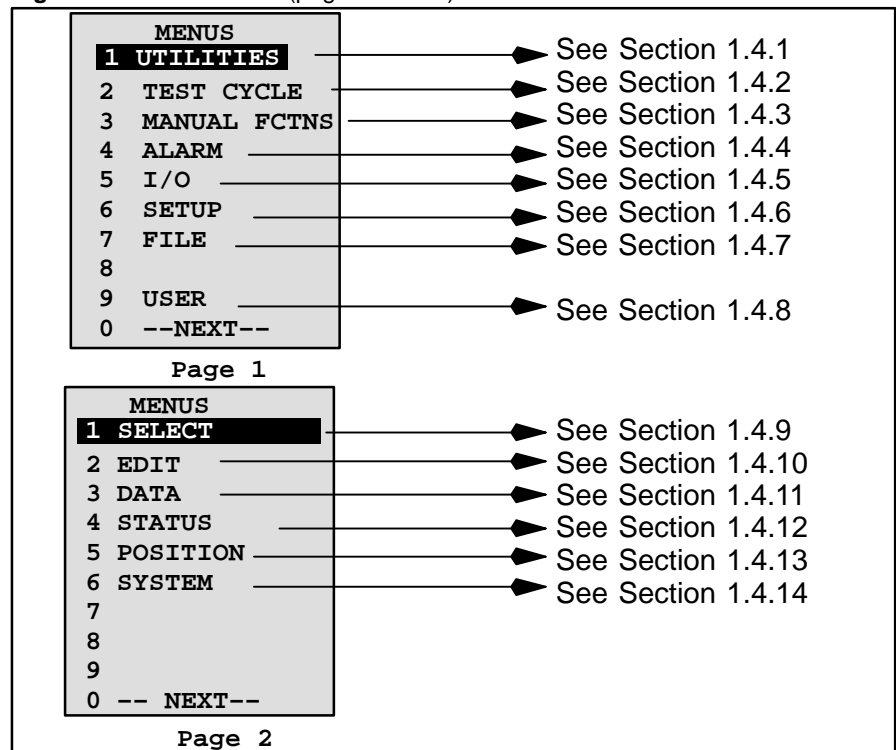
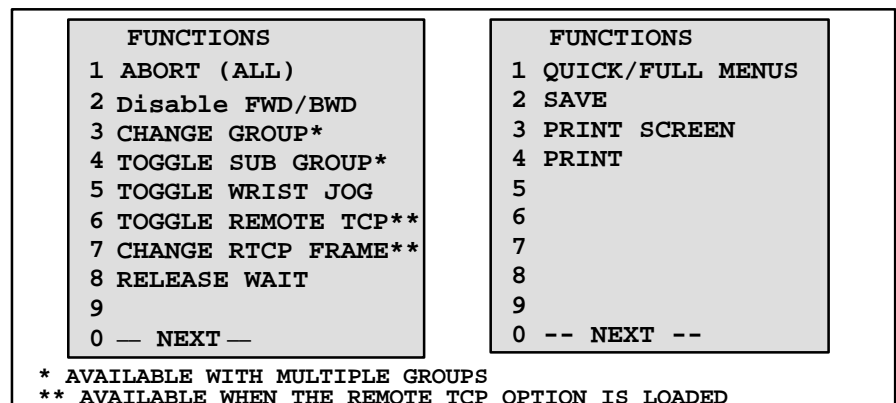


Figure 1-34. FCTN Menu (pages 1 and 2)



1.4.1 UTILITIES Menu Map

Figure 1-35. UTILITIES Menu Map

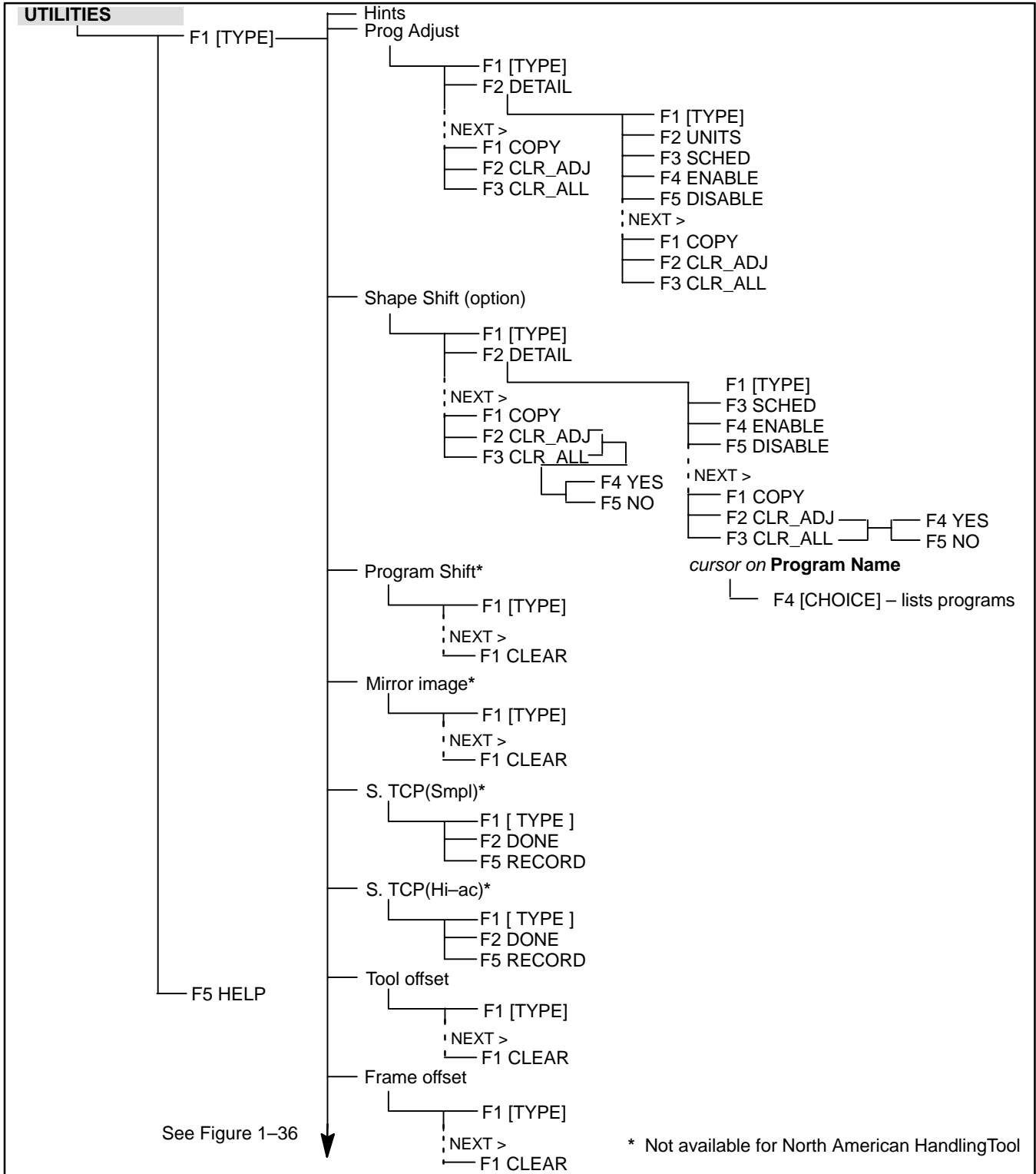
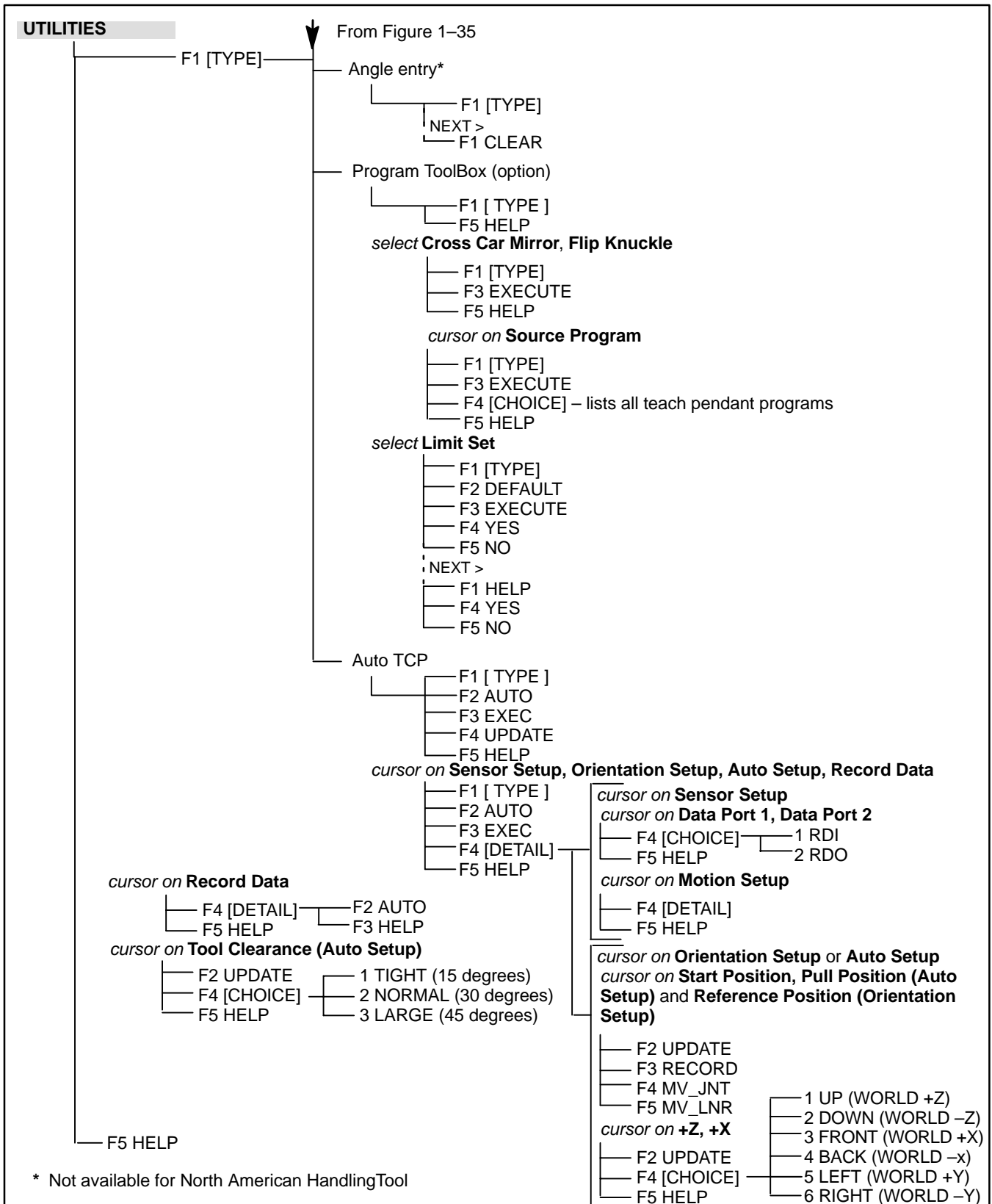


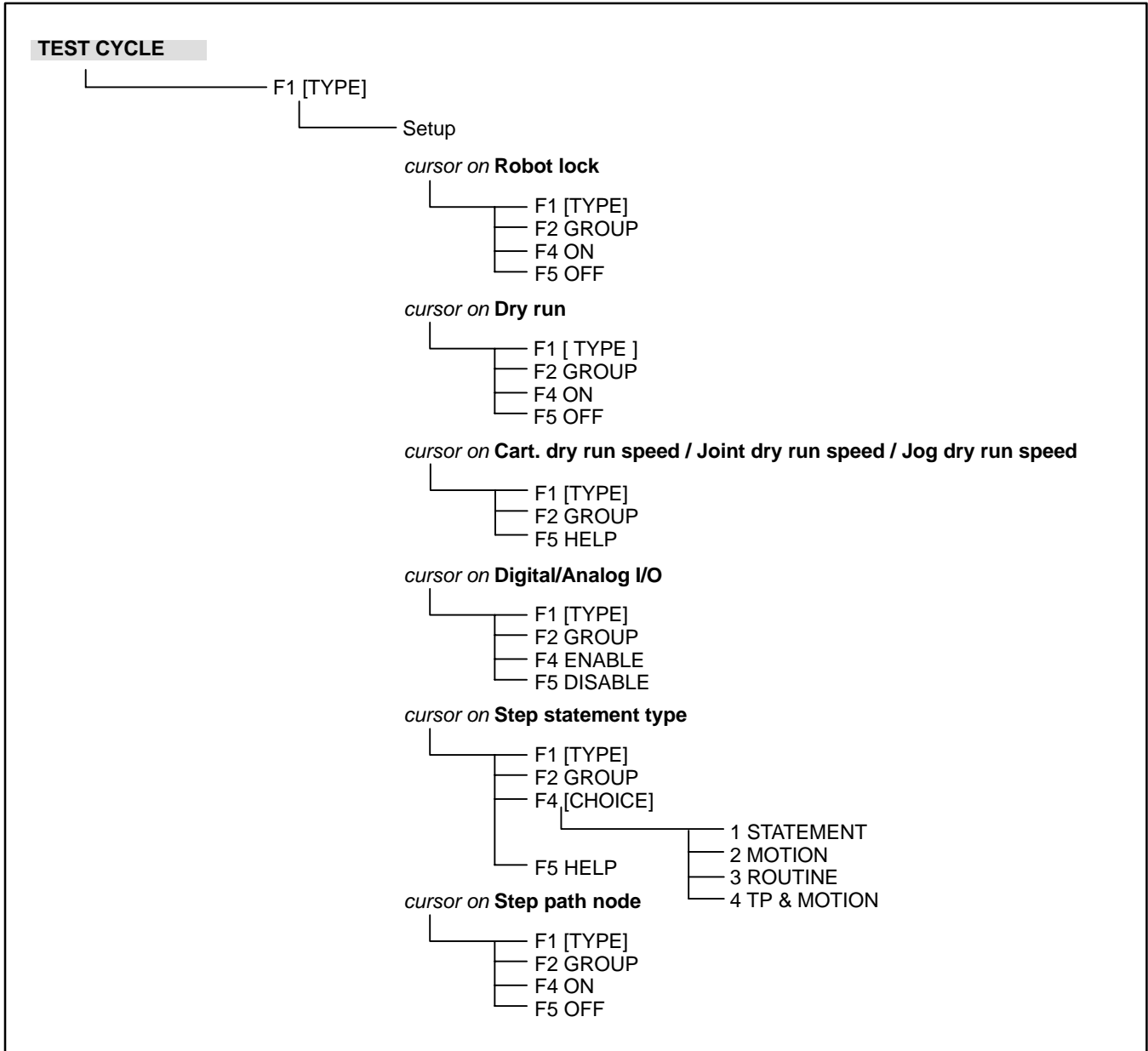
Figure 1-36. UTILITIES Menu Map (continued)



* Not available for North American HandlingTool

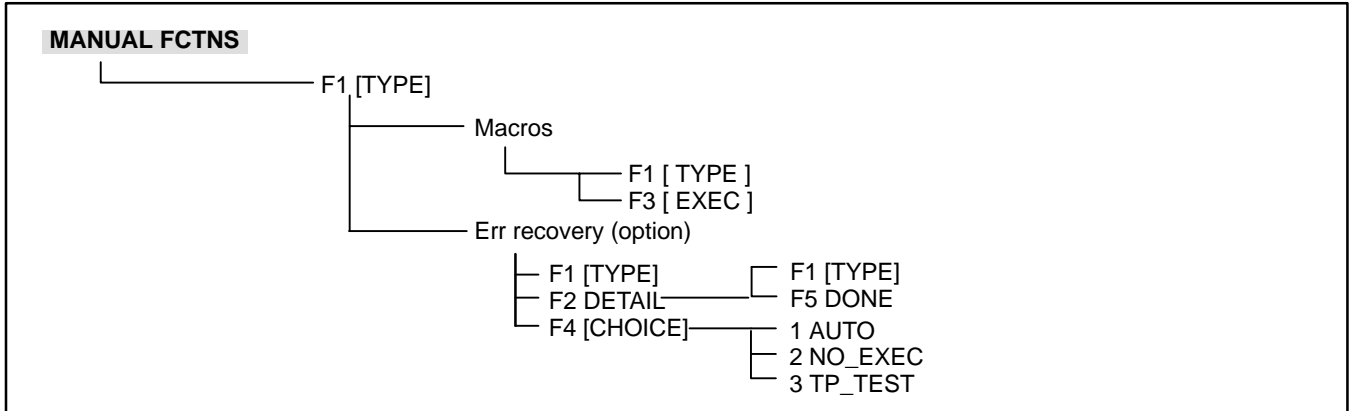
1.4.2 TEST CYCLE Menu Map

Figure 1-37. TEST CYCLE Menu Map



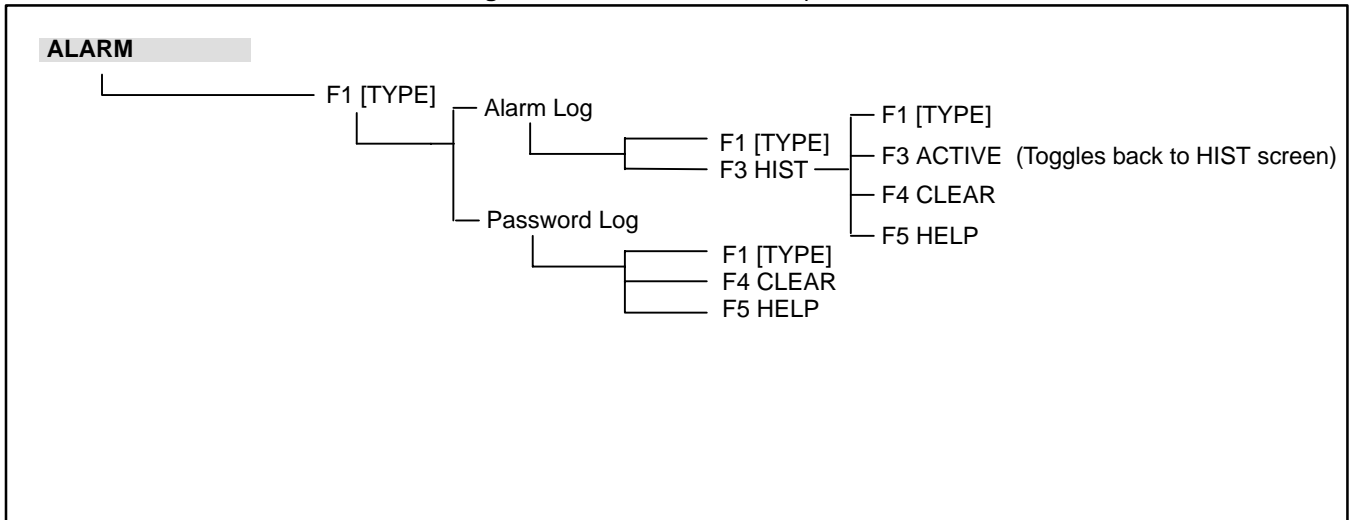
1.4.3 MANUAL FCTNS Menu Map

Figure 1-38. MANUAL FCTNS Menu Map



1.4.4 ALARM Menu Map

Figure 1-39. ALARM Menu Map



1.4.5 I/O Menu Map

Figure 1-40. I/O Menu Map

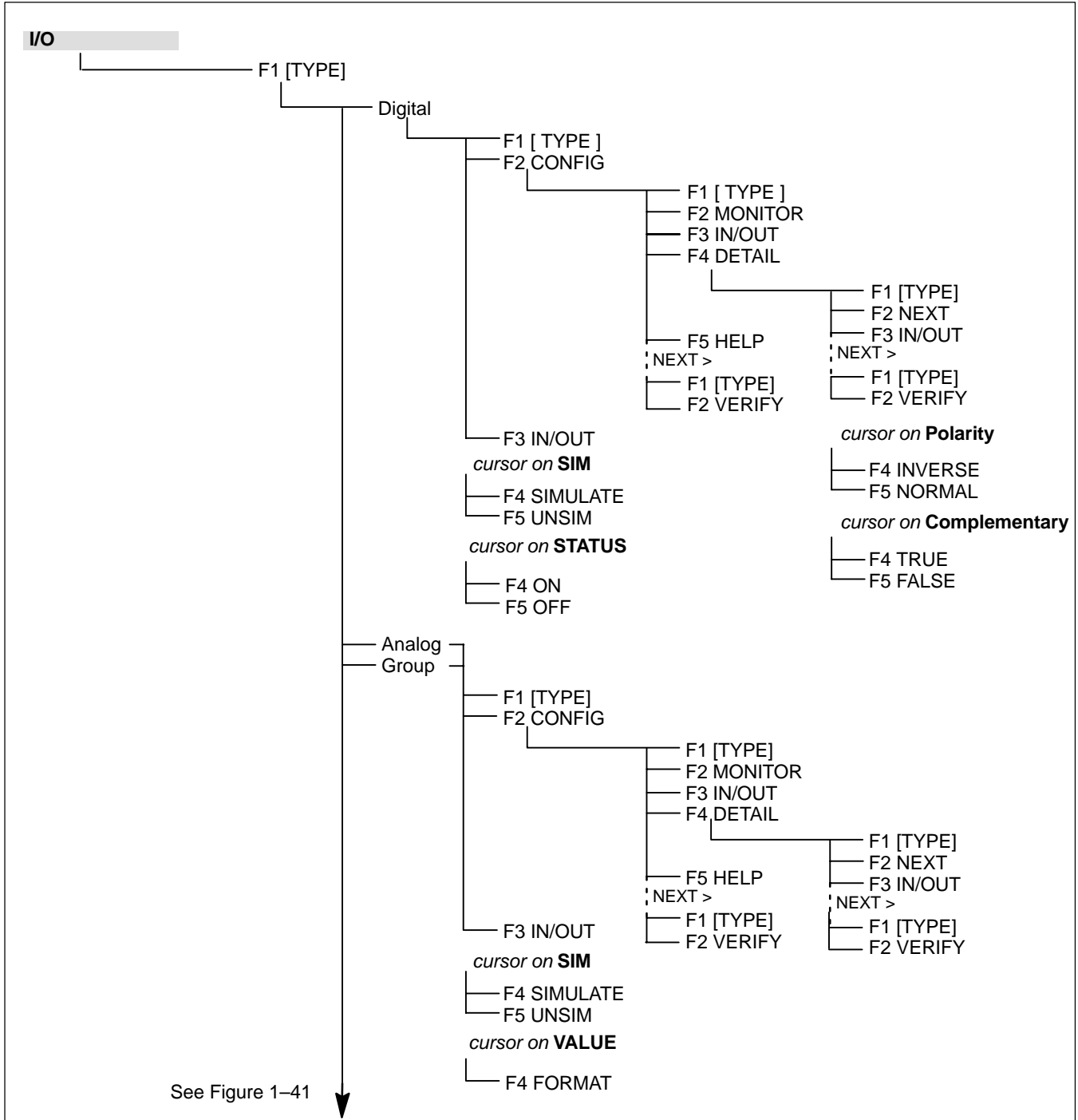


Figure 1-41. I/O Menu Map (Continued)

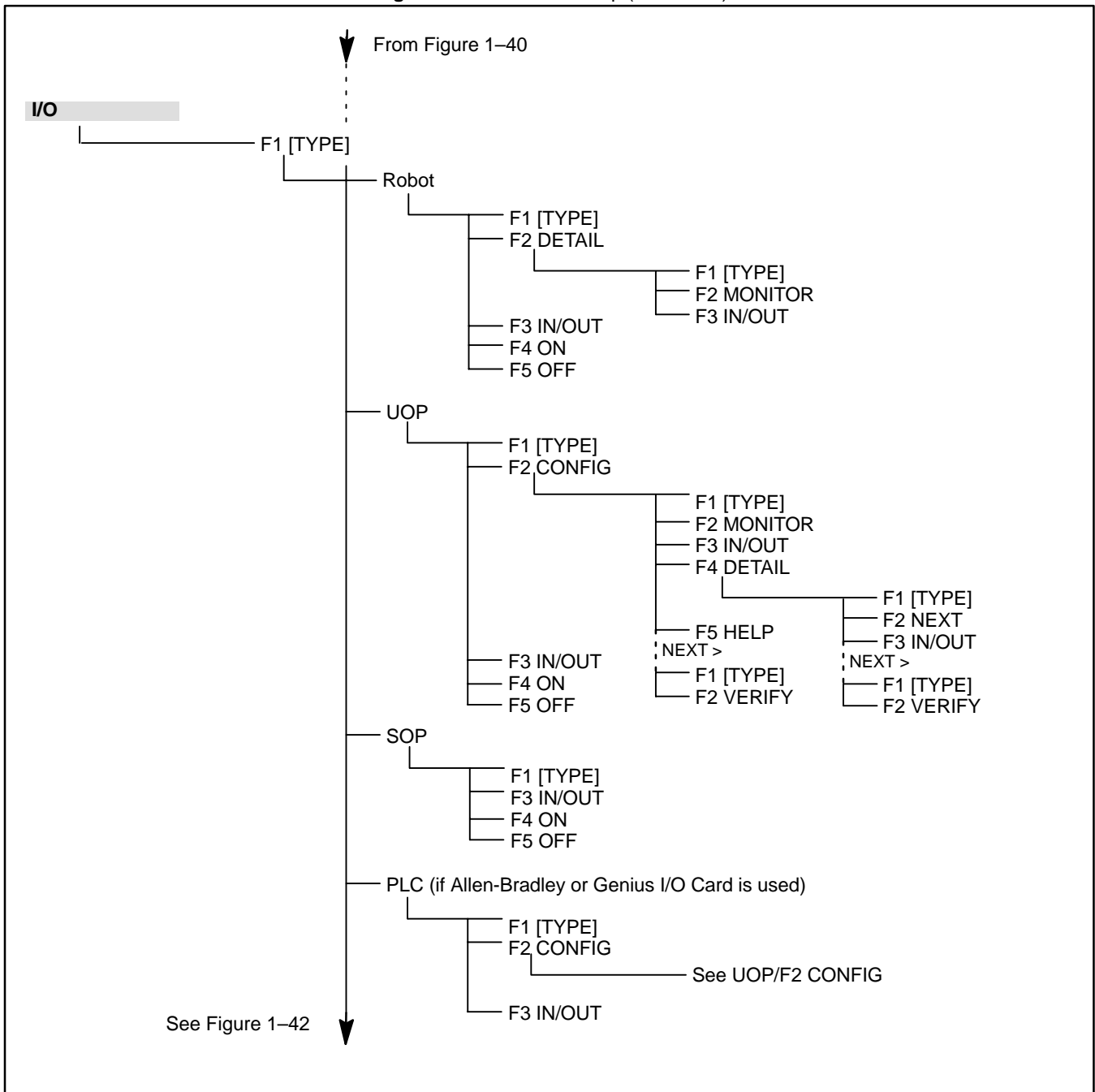
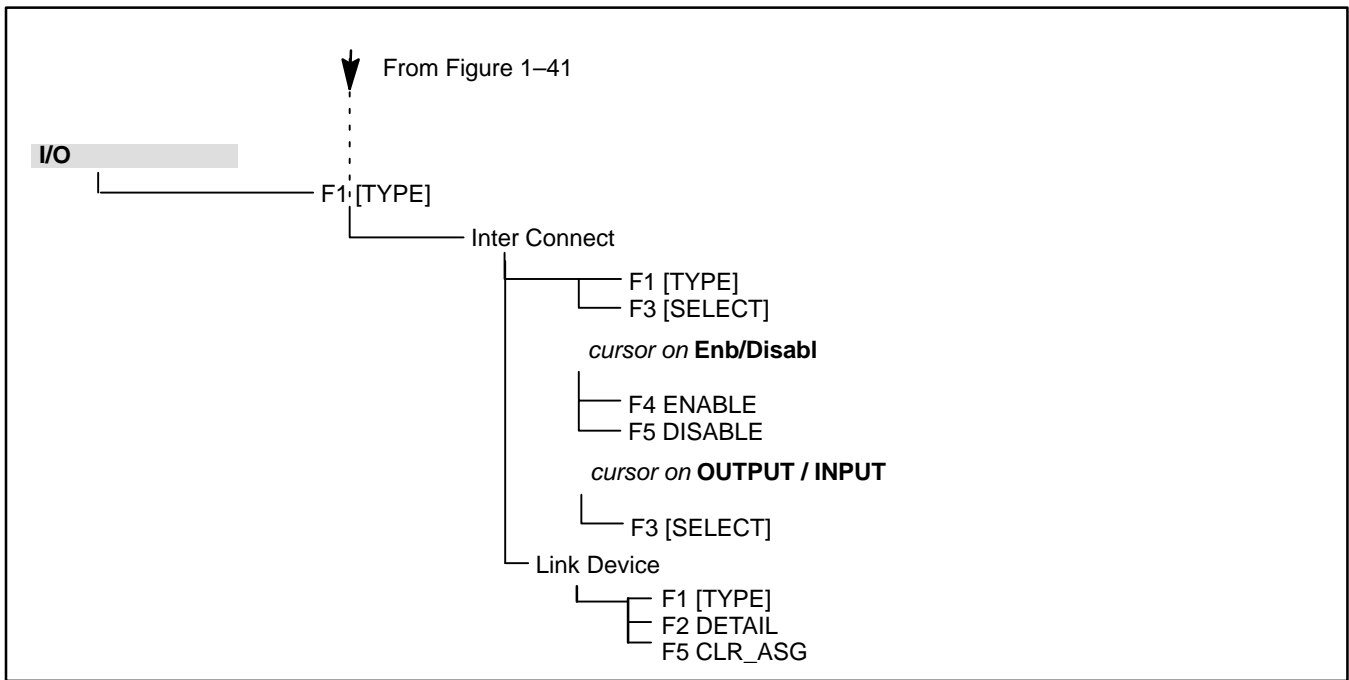
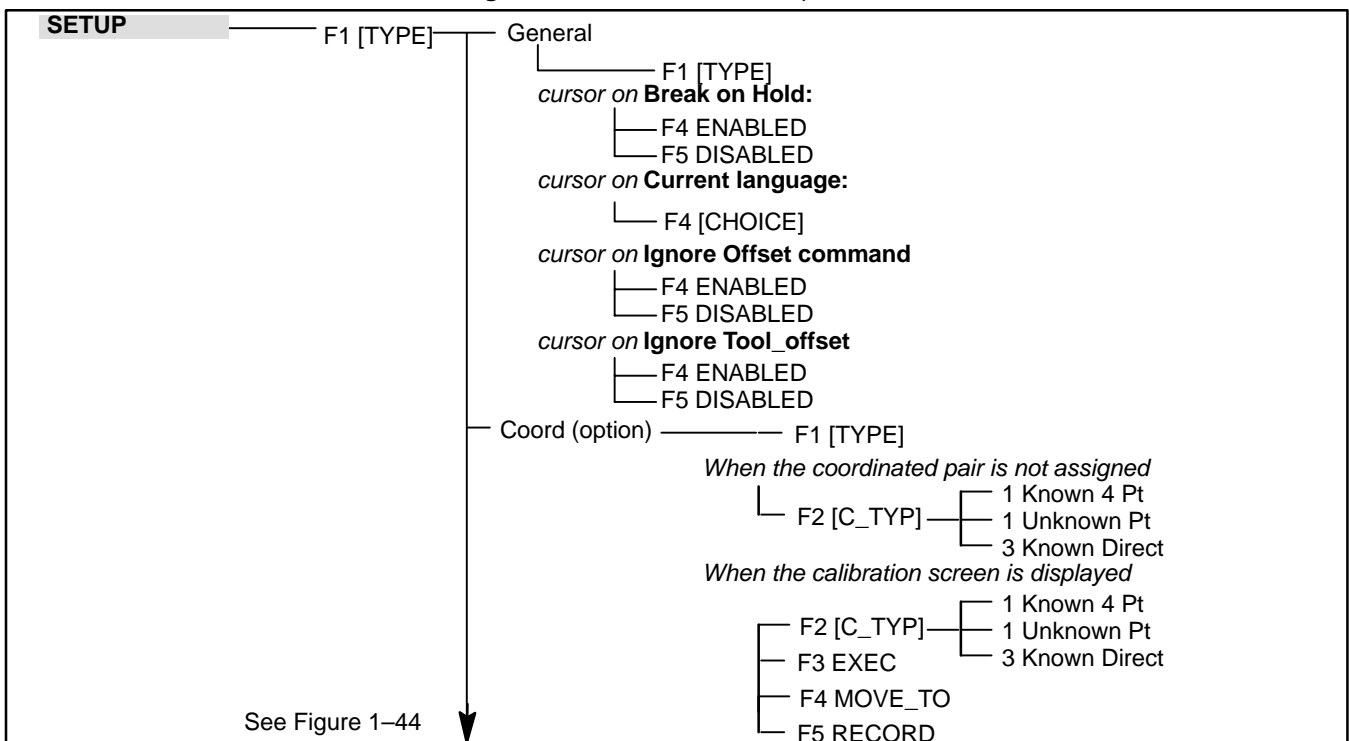


Figure 1-42. I/O Menu Map (Continued)



1.4.6 SETUP Menu Map

Figure 1-43. SETUP Menu Map



See Figure 1-44

Figure 1-44. SETUP Menu Map (Continued, Page 2)

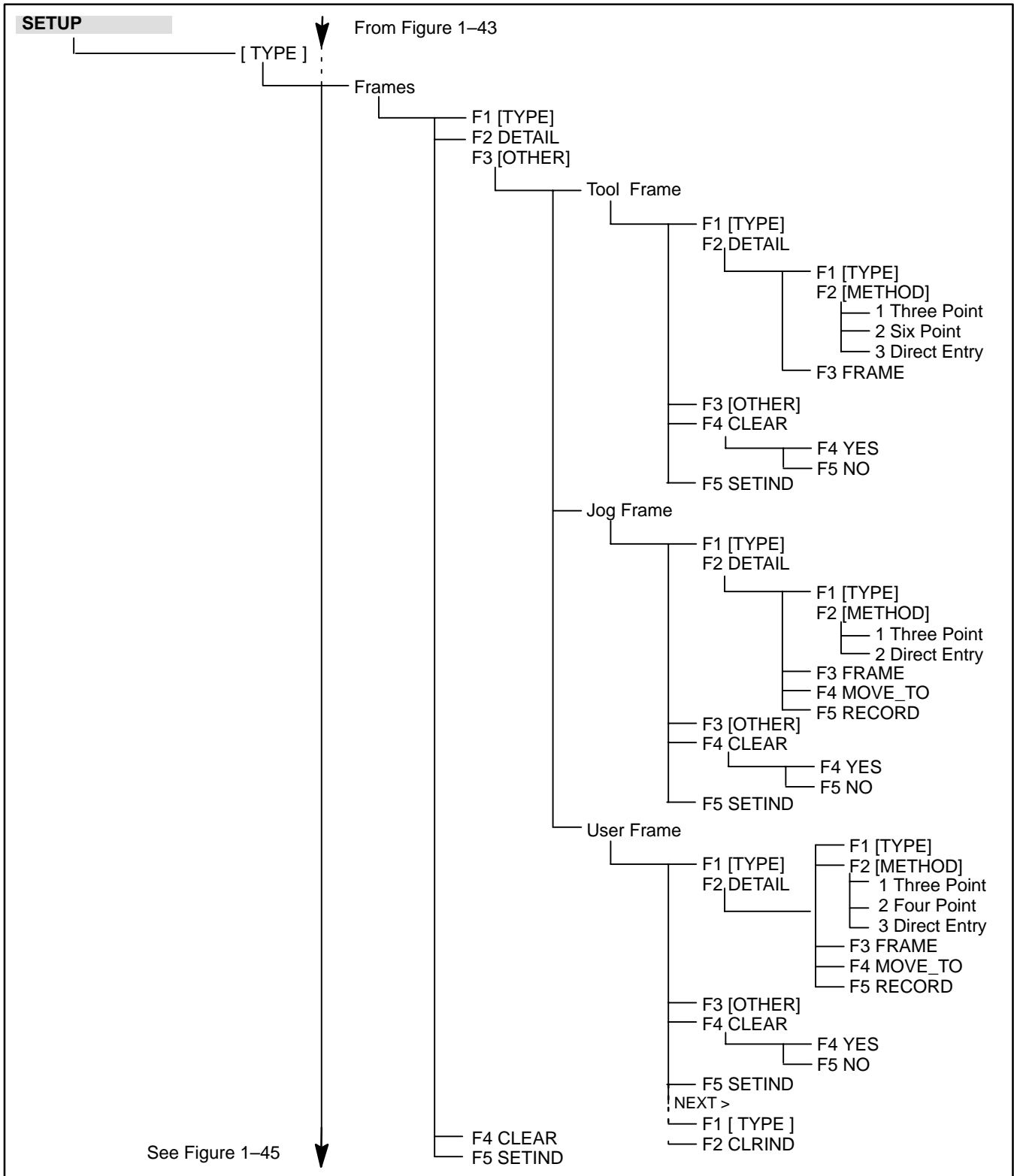


Figure 1-45. SETUP Menu Map (Continued, Page 3)

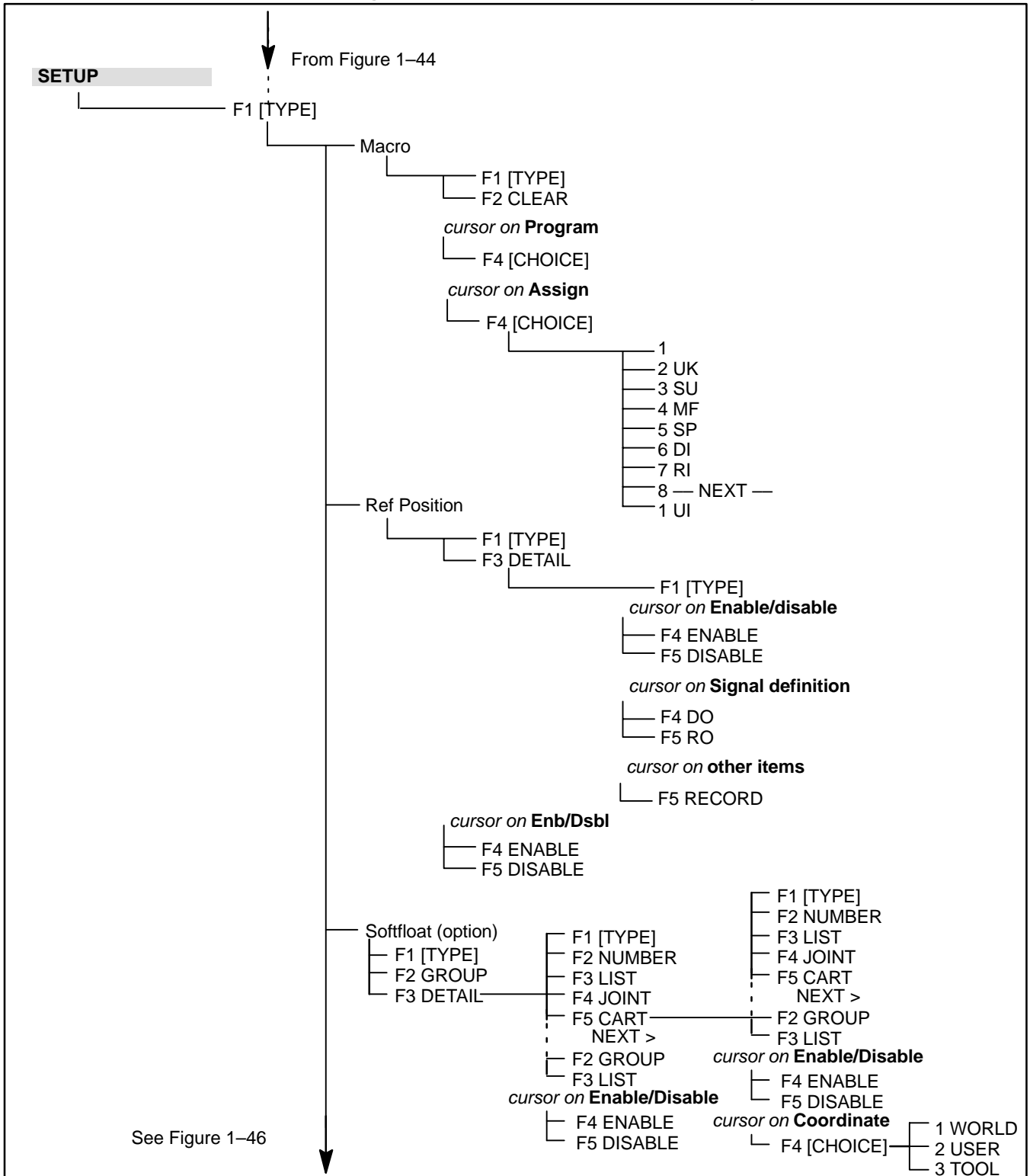


Figure 1-46. SETUP Menu Map (Continued, Page 4)

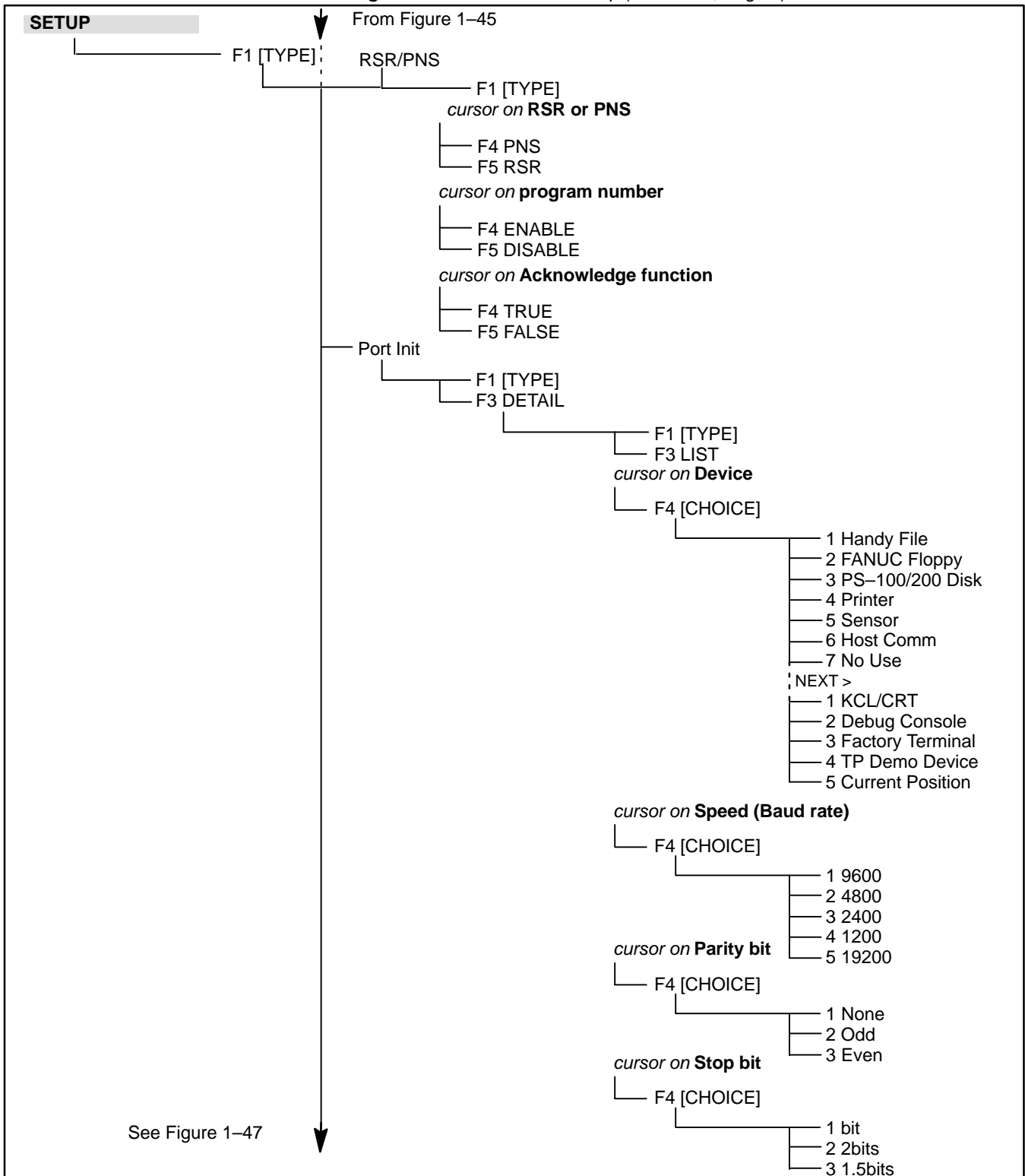
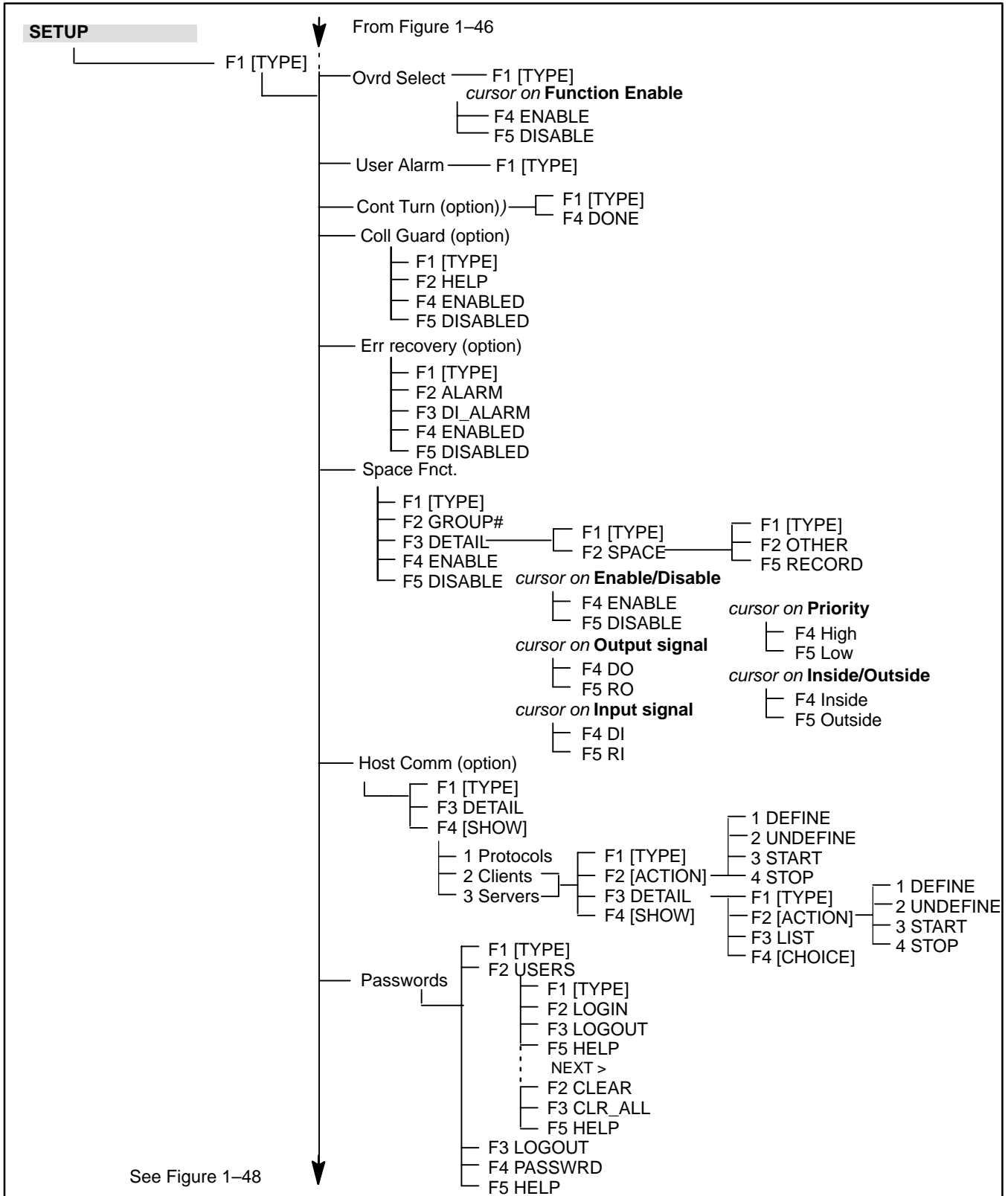
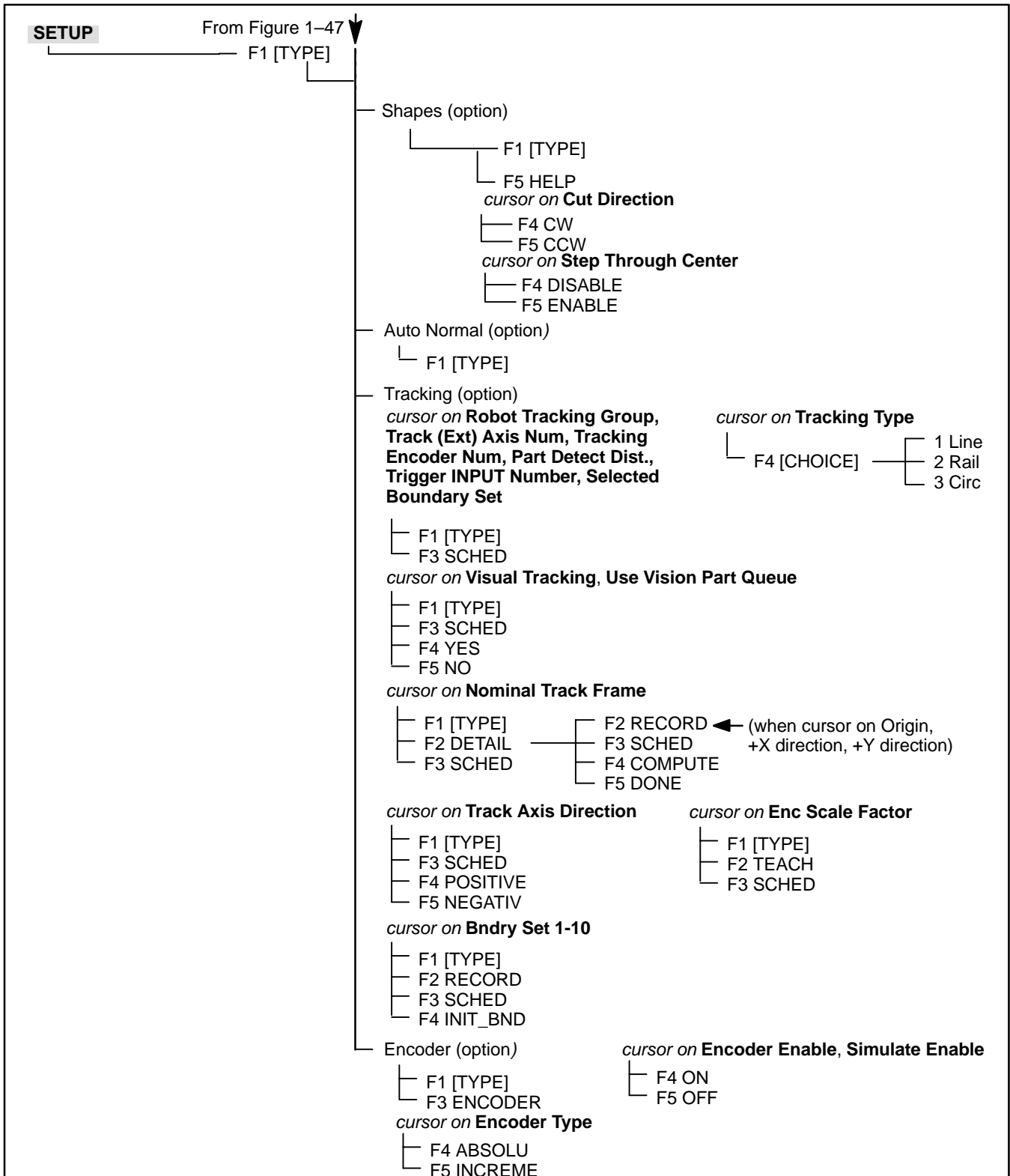


Figure 1-47. SETUP Menu Map (Continued, Page 5)



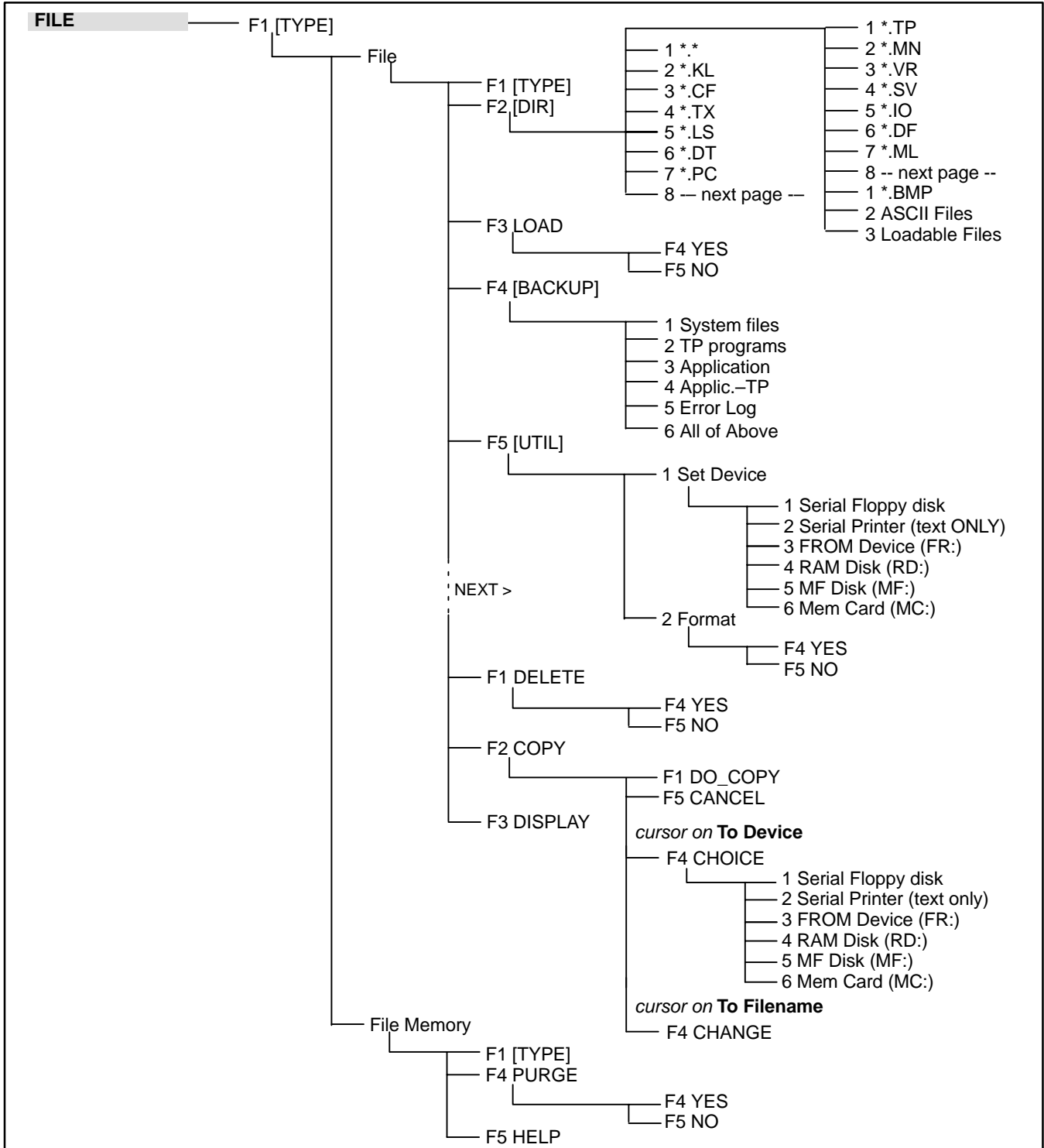
See Figure 1-48

Figure 1-48. SETUP Menu Map (Continued, Page 6)



1.4.7 FILE Menu Map

Figure 1-49. FILE Menu Map

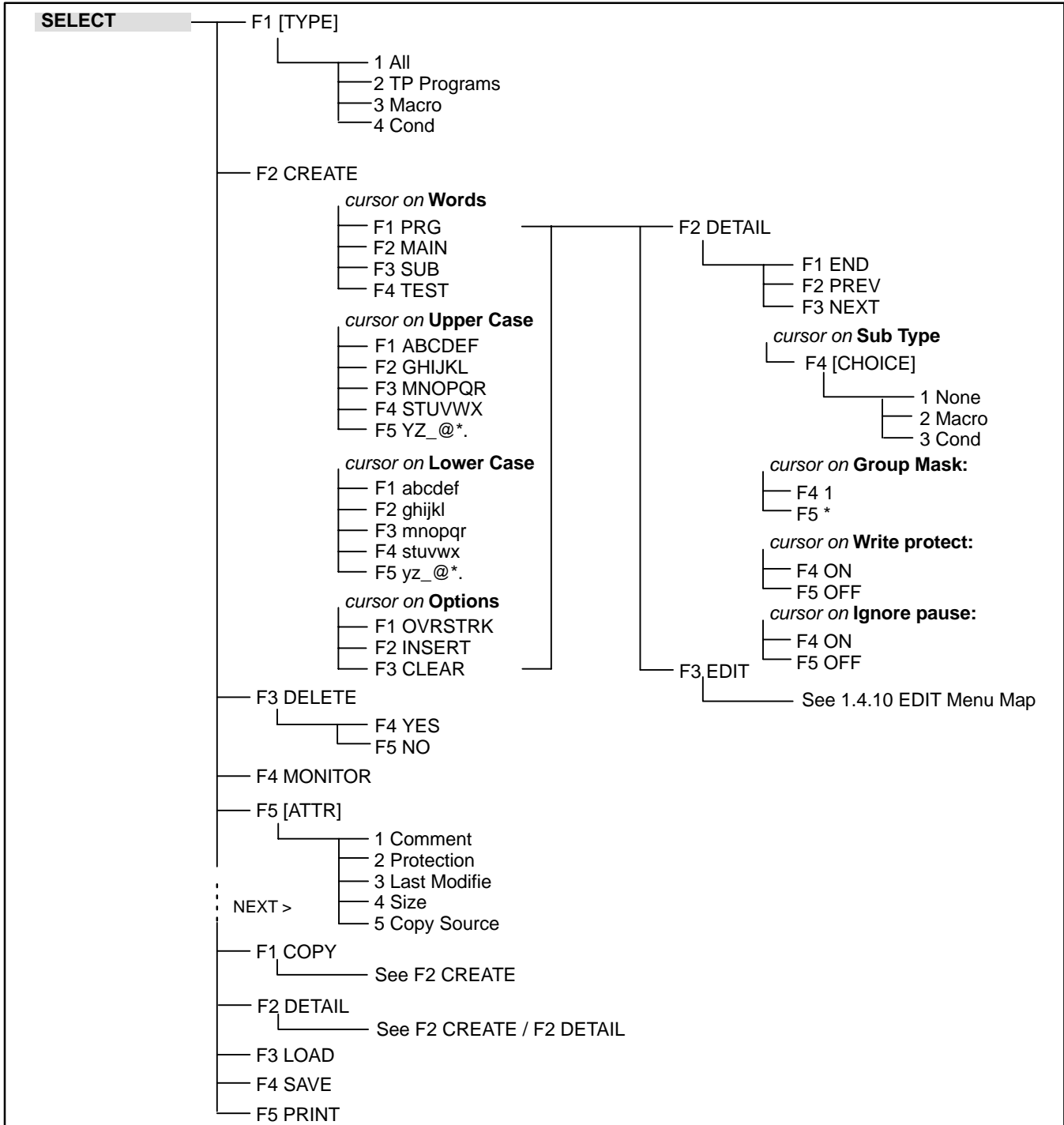


1.4.8 USER Menu Map

The items on the User Menu are user-defined. Refer to Section 6.12.6 (Message Instruction).

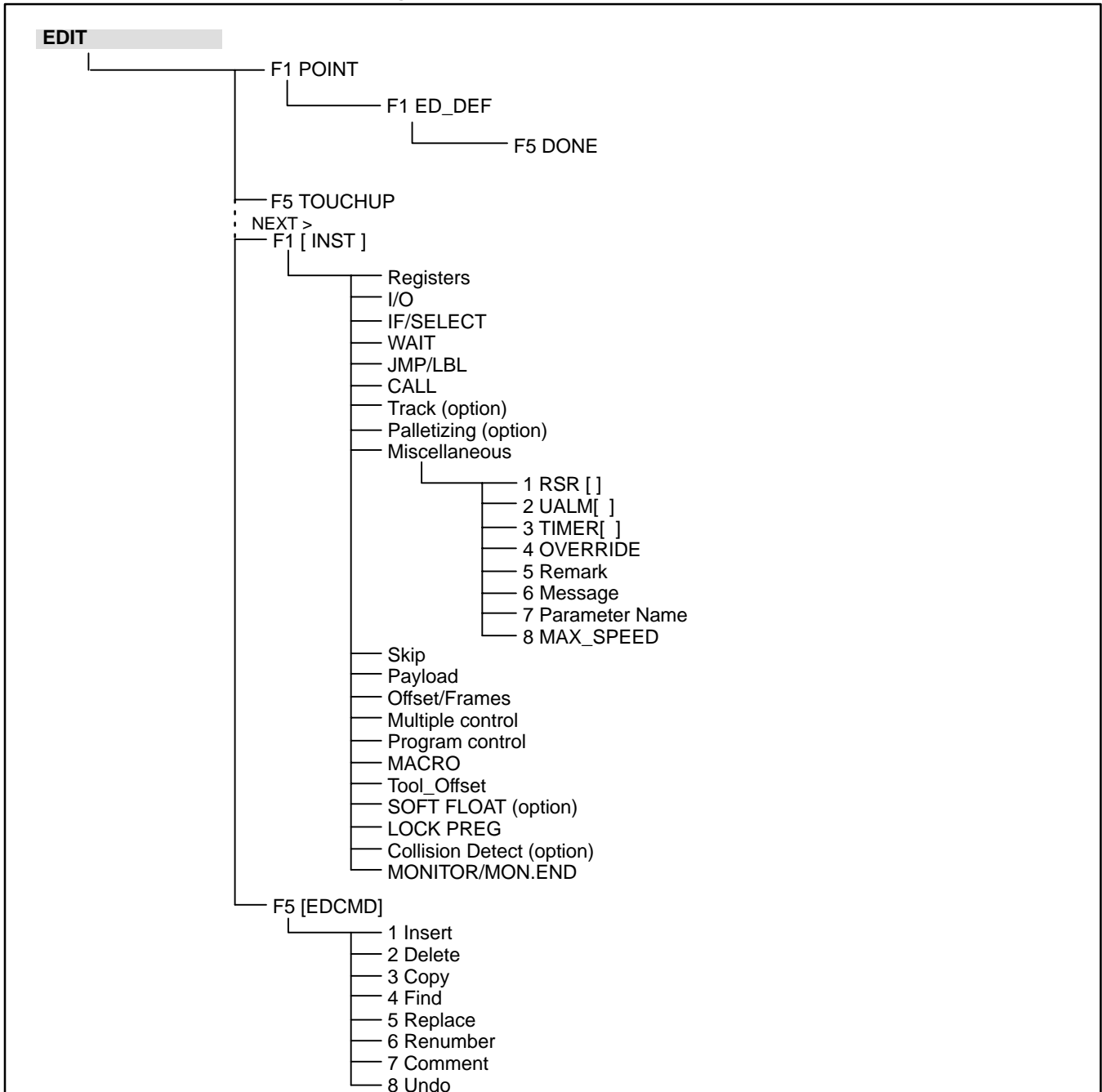
1.4.9 SELECT Menu Map

Figure 1-50. SELECT Menu Map



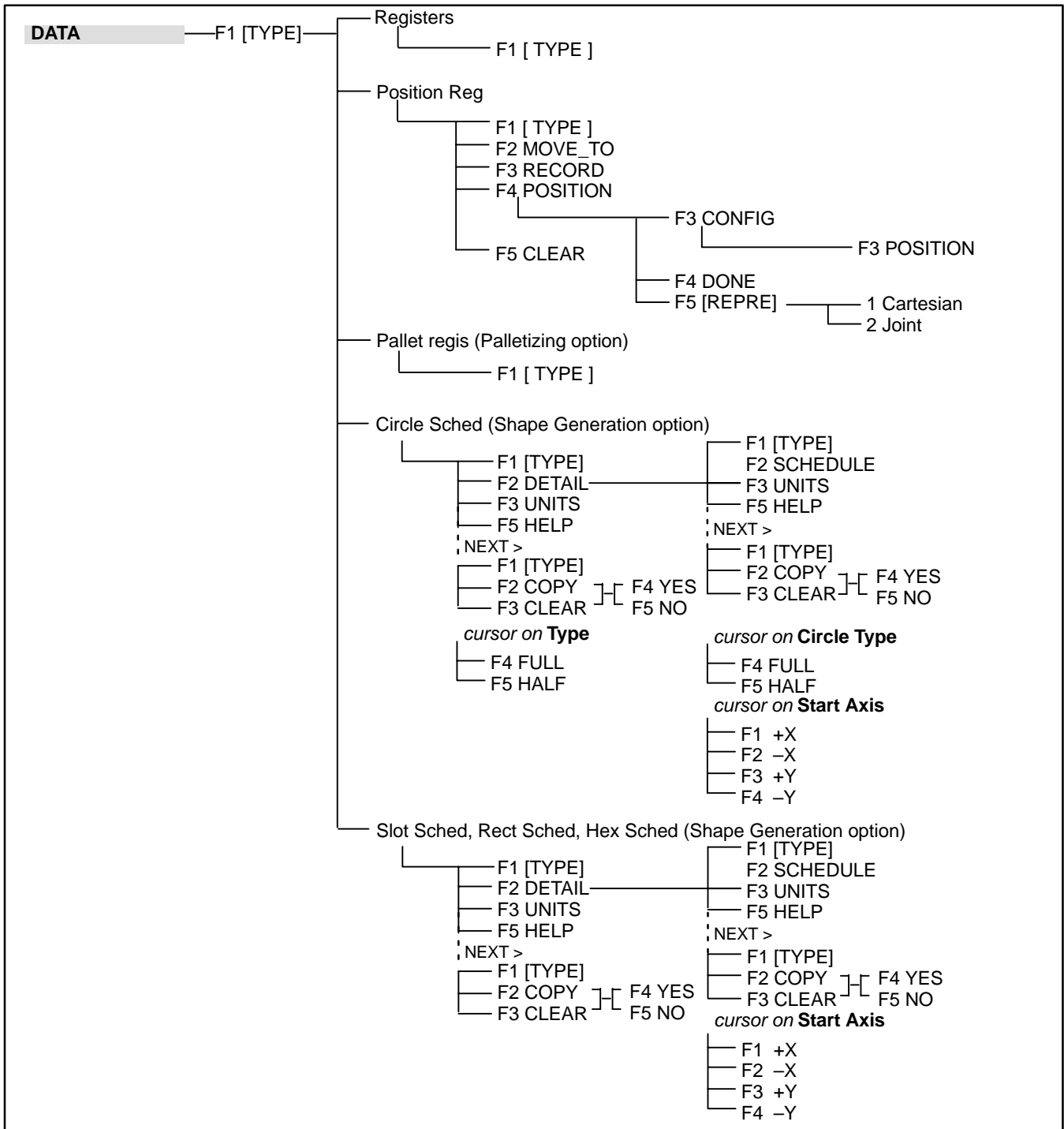
1.4.10 EDIT Menu Map

Figure 1-51. EDIT Menu Map



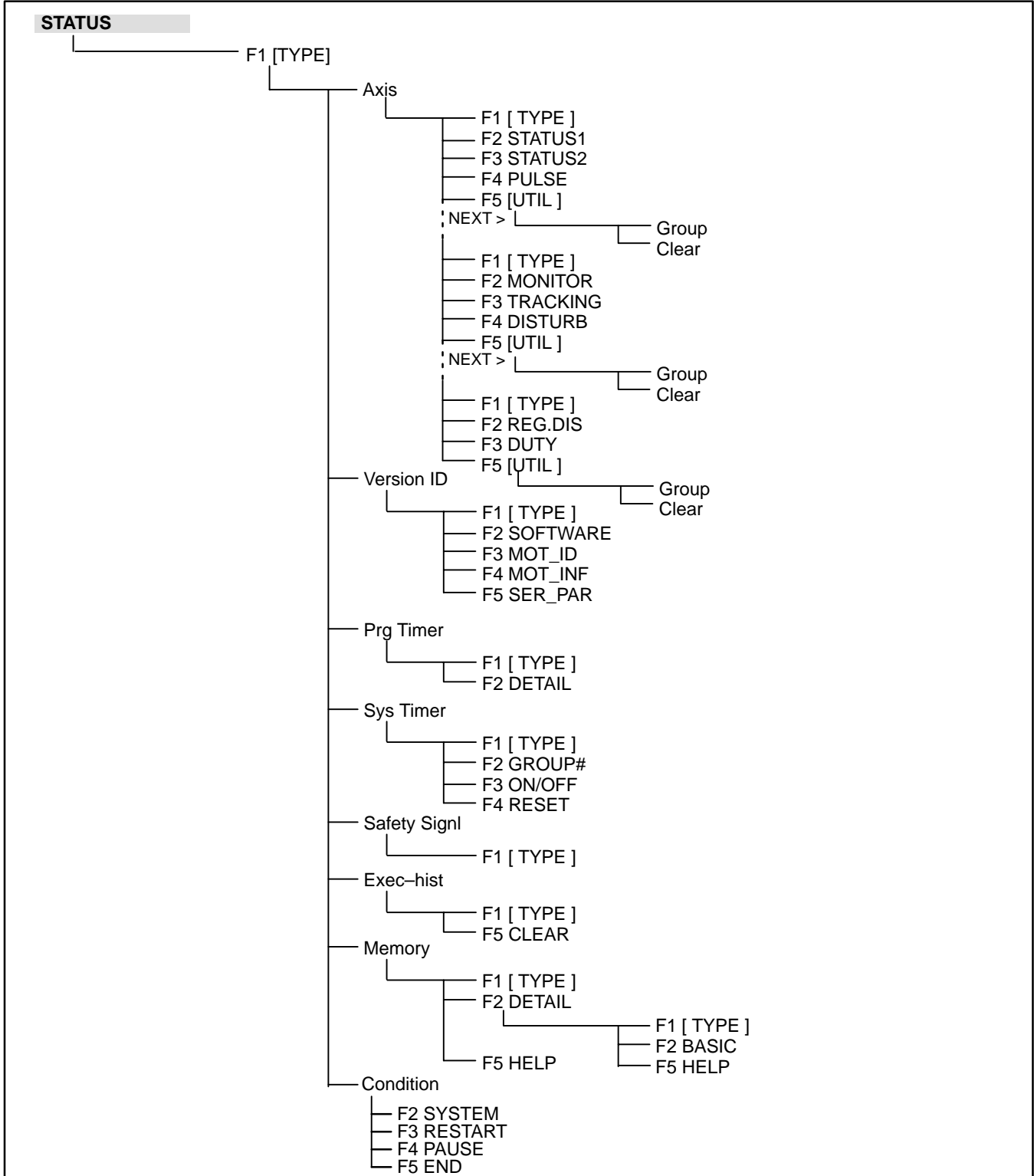
1.4.11 DATA Menu Map

Figure 1-52. DATA Menu Map



1.4.12 STATUS Menu Map

Figure 1-53. STATUS Menu Map



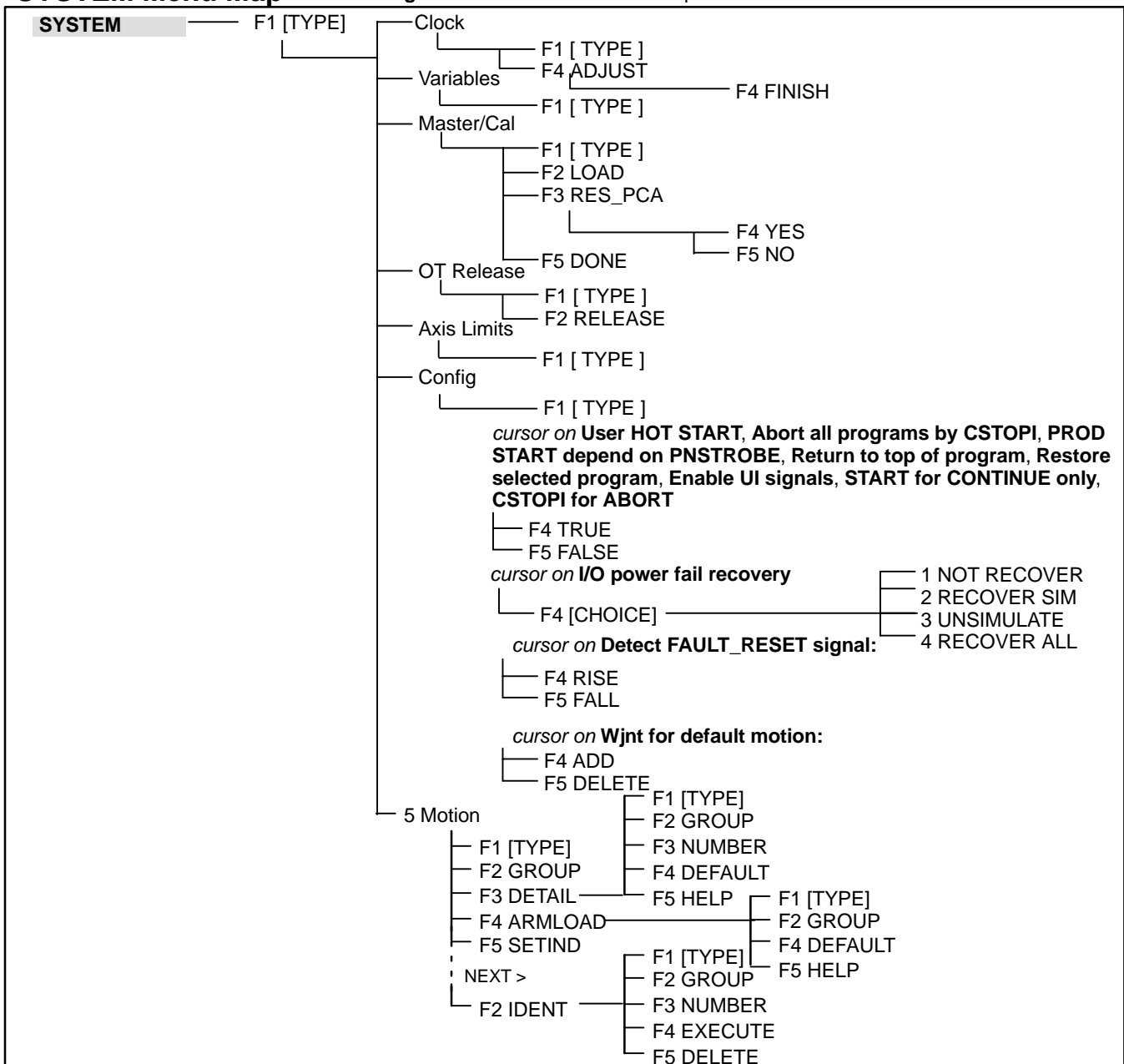
1.4.13 POSITION Menu Map

Figure 1-54. POSITION Menu Map



1.4.14 SYSTEM Menu Map

Figure 1-55. SYSTEM Menu Map



2 TURNING ON AND JOGGING THE ROBOT

2

TURNING ON AND JOGGING THE ROBOT

Topics In This Chapter		Page
Turning On the Robot	Turning on the robot provides power to the robot and controller.	2-2
Jogging the Robot	Jogging is moving the robot axes by pressing keys on the teach pendant.	2-5
	• Jogging when Activation from the Teach Pendant is Inhibited	2-5
	• Jog Speed	2-6
	• Coordinate Systems	2-7
	• Wrist Jogging	2-9
	• Remote TCP Jogging (option)	2-9
	• Motion Groups	2-10
	• Extended Axes and Sub-Groups	2-10
	• Jog Menu	2-15

Before you can create a program and run production you must first know how to turn on and jog the robot. Turning on the robot provides power to the robot and controller. Jogging is moving the robot by pressing keys on the teach pendant. This chapter contains information and procedures to turn on and off and to jog the robot.

2.1 TURNING ON AND TURNING OFF THE ROBOT

Turning on the robot provides power to the robot and controller and does the following:

- Initializes changes to system variables
- Initializes changes to I/O setup
- Displays the UTILITIES Hints screen

Refer to Appendix C for more information on startup methods.

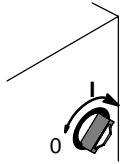
Use Procedure 2-1 to turn on the robot. Use Procedure 2-2 to turn off the robot.

 **CAUTION**

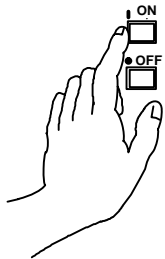
Your plant might require additional inspections before turning on power to the robot. To help ensure safe operation, become familiar with the guidelines for your installation before you turn on the robot.

Procedure 2-1 Turning On the Robot

Condition
Step



CIRCUIT
BREAKER
I-SIZE/B-SIZE
CONTROLLER



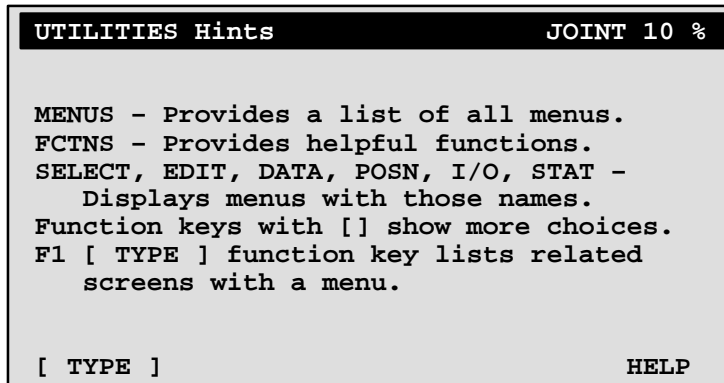
- All personnel and unnecessary equipment are out of the workcell.
- 1 Visually inspect the robot, controller, workcell, and the surrounding area. During the inspection make sure all safeguards are in place and the work envelope is clear of personnel.
- 2 Turn the circuit breaker on the operator panel to ON.



WARNING

DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.

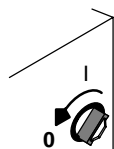
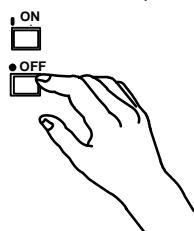
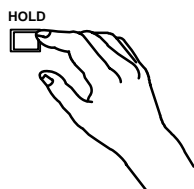
- 3 Press the ON button on the operator panel.
 - **On the operator panel**, the ON button will be illuminated, indicating robot power is on.
 - **On the teach pendant screen**, various messages will appear on the screen for a brief period, and then you will see the UTILITY Hints screen. See the following screen for an example.



For North American HandlingTool, you will see a screen similar to the following.



Procedure 2-2 Turning Off the Robot



CIRCUIT BREAKER
SWITCH
i-SIZE/B-SIZE
CONTROLLER

- Step**
- 1** If a program is running or if the robot is moving, press the HOLD button on the operator panel or press the HOLD button on the teach pendant or send the HOLD signal from a UOP.
 - 2** Perform any shutdown procedures specific to your installation.
 - 3** Press the OFF button on the operator panel.
 - 4** Turn the circuit breaker to OFF when performing maintenance on the robot or controller.

**WARNING**

When you open the controller cabinet you must disconnect power at the supply. Turning OFF the power at the controller does not protect you from the live power at the switch.

2.2 JOGGING THE ROBOT

Jogging is moving the robot axes by pressing keys on the teach pendant.

Six items affect the way the robot jogs and the axes that move while jogging. The four items that affect the way the robot jogs are

- **Jog speed** – How fast the robot moves when jogging
- **Coordinate system** – The way the robot moves when jogging
- **Minor Axis Wrist Jog** – How the wrist axes will jog
- **Remote TCP jogging** – Whether the tool is fixed in the workcell

The two items that affect the axes that move while jogging are

- **Motion Groups** – Which motion group is selected
- **Extended axes and motion sub-groups** – Which extended axes or sub-group is selected

2.2.1 Jogging when Activation from the Teach Pendant is Inhibited

This function disables jogging unless the robot is prevented from being activated using the teach pendant (TP). In other words, this function can specify that jogging is enabled only when the robot cannot be activated using the TP.

This function is enabled and disabled using system variable `$SCR.$TPMOTNENABL`.

To enable this function (to allow jog feed only when activation using the TP is inhibited), change the value of the system variable from 0 to 1 (or 2 to 3) on the SYSTEM Variables menu.

The table below lists the values of system variable `$SCR.$TPMOTNENABL`, whether this function is enabled, and whether jogging is enabled. Refer to Table 2-1.

Table 2-1. Relationship of Jogging to `$SCR.$TPMOTNENABL`

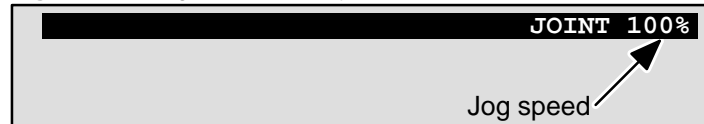
<code>\$SCR.\$TPMOTNENABL</code>	Activation using the TP	Jogging
0	Enabled	Enabled
1	Enabled	Disabled
2	Disabled	Enabled
3	Disabled	Enabled

By default, this function is disabled.

2.2.2 Jog Speed

The jog speed is a percentage of the maximum speed at which you can jog the robot. The current jog speed is displayed in the right corner of every teach pendant screen, as shown in Figure 2-1.

Figure 2-1. Jog Speed Display



A jog speed of 100% indicates that the robot will move with the maximum possible jog speed. The maximum possible jog speed varies depending on the type of robot. A jog speed of VFINE indicates that the robot will move in incremental steps with the minimum possible jog speed.

Table 2-2 lists all the possible values of the jog speed.

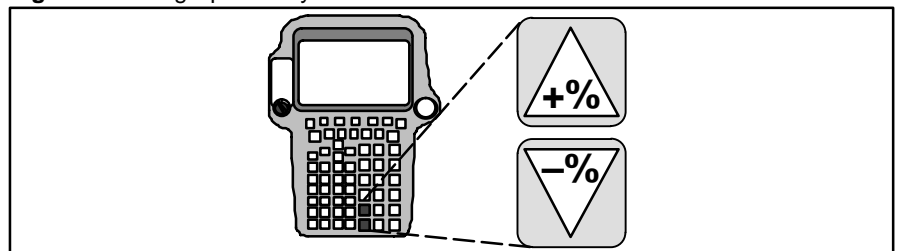
NOTE When using FINE and VFINE speed values, the robot moves one step at a time. You must release the jog key and press it again to move the robot again.

Table 2-2. Jog Speed Values

SPEED VALUES	JOINT	CARTESIAN
100, 95, 90, 85, ... 20, 15, 10, 5, 4, 3, 2, 1	% of jog speed	% of jog speed
FINE (incremental steps)	Approximately 0.001 degrees	Approximately 0.023 mm
VFINE (incremental steps)	Approximately 0.0001 degrees	Approximately 0.002 mm

The jog speed keys on the teach pendant are used to increment or decrement the jog speed. The SHIFT key combined with a jog speed key causes the jog speed to be changed between 100, 50, 5, FINE, and VFINE. Figure 2-2 shows the jog speed keys.

Figure 2-2. Jog Speed Keys



Set the jog speed to a value that is appropriate for the conditions in the workcell, the type of jogging the robot is doing, and your own experience in jogging a robot. Use a slow jog speed until you are familiar with the robot. The slower the jog speed, the more control you have over robot motion.

NOTE JOG SPEED increments only when the COORD/JOG SPEED window on the teach pendant is displayed.

- Press the COORD or % JOG SPEED key to display the COORD/JOG SPEED screen.
- Press the % JOG SPEED keys to change the JOG SPEED values.

2.2.3 Coordinate Systems

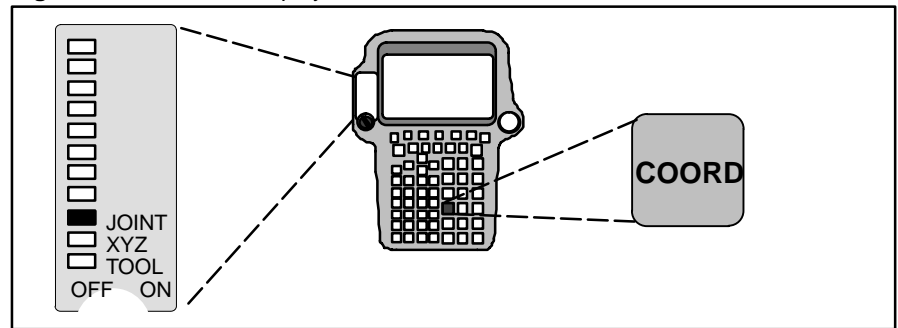
In jogging, a coordinate system defines how the robot will move. There are three coordinate systems:

- JOINT
- XYZ – includes WORLD, JGFRM, and USER
- TOOL

You change the coordinate system by pressing the COORD key on the teach pendant, shown in Figure 2-3. The coordinate system you choose is displayed in the upper right hand corner of the teach pendant screen, and on the teach pendant LEDs.

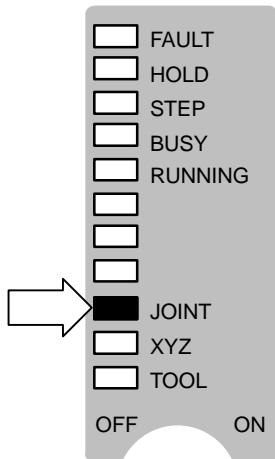
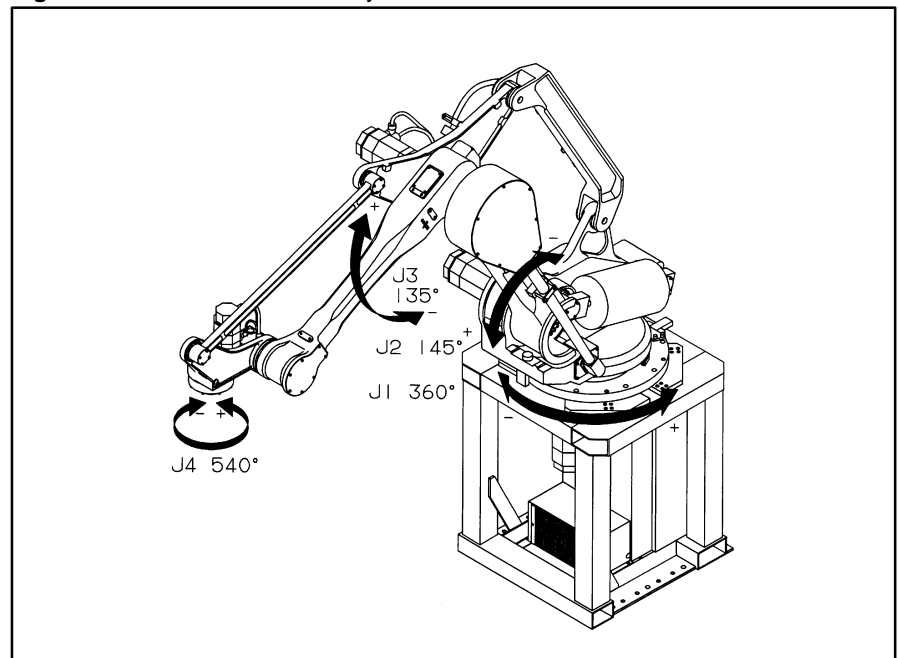
Press the COORD key to change the coordinate system.

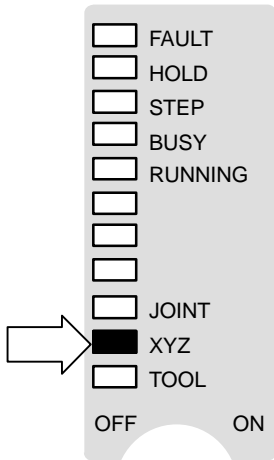
Figure 2-3. COORD Display



- **JOINT coordinate system** – Moves the individual axes of the robot. See Figure 2-4.

Figure 2-4. JOINT Coordinate System

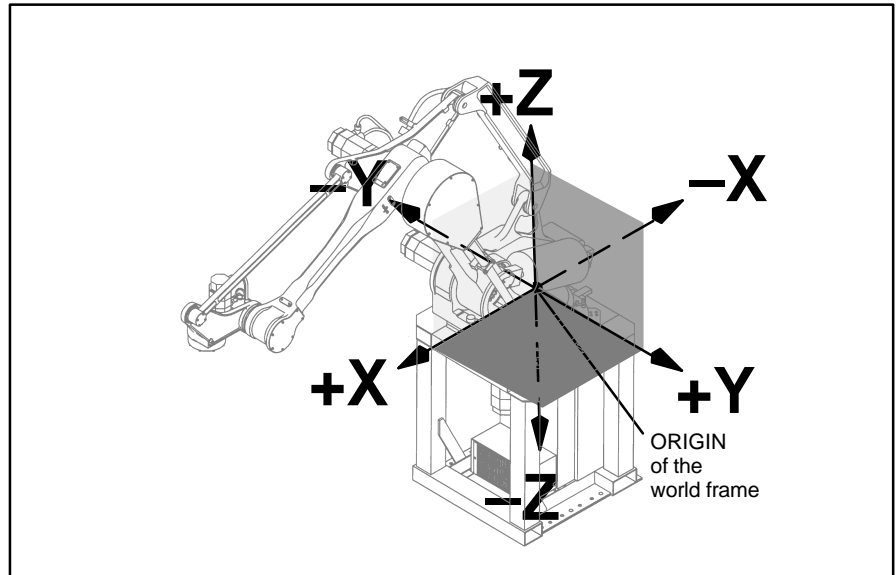




- XYZ coordinate system** – Moves the face plate of the robot in the x, y, or z directions and about the x, y, or z (w, p, or r) axes. XYZ coordinate systems are WORLD and JGFRM (jog frame).

You can jog the robot using either the WORLD or JGFRM (jog frame) coordinate systems. See Figure 2-5.

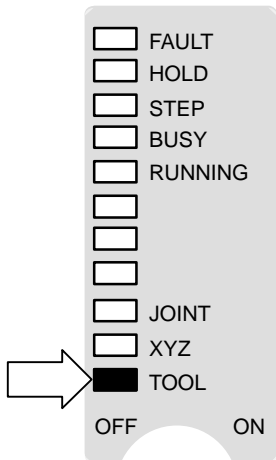
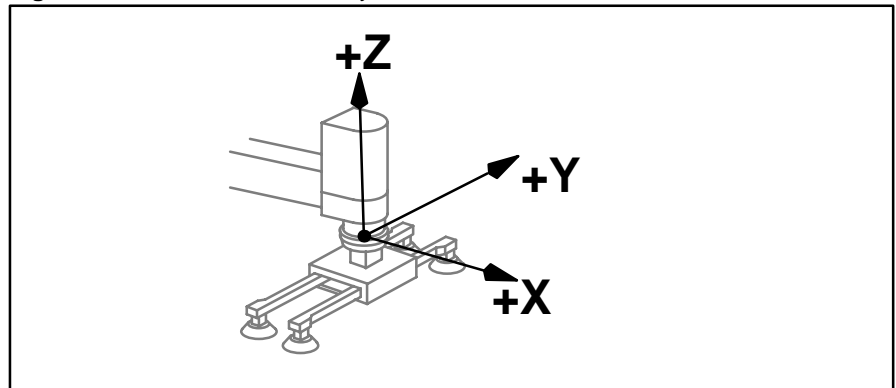
Figure 2-5. XYZ Coordinate System



- TOOL coordinate system** – Moves the robot TCP in the x, y, or z direction, and rotates about x (w), y (p), and z (r) in the selected tool frame.

See Figure 2-6.

Figure 2-6. TOOL Coordinate System



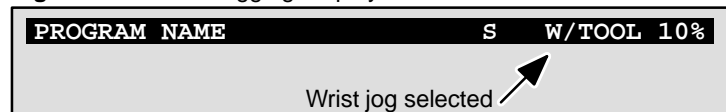
2.2.4 Wrist Jogging

The wrist jog function allows you to control how the robot axes will jog when you are using a Cartesian coordinate system, such as WORLD or TOOL. Wrist jogging does not affect x, y, and z jogging, it affects only orientation jogging.

When you jog a wrist axis using wrist jog, the other wrist axes will remain stationary and the rest of the robot axes will move to accommodate the movement of the wrist axes to maintain a fixed TCP location.

You select wrist jog using the FCTN menu. When you select wrist jog, “W” appears next to the coordinate system name displayed on the teach pendant screen. See Figure 2-7. Use Procedure 2-3 to select wrist jog and jog the axes.

Figure 2-7. Wrist Jogging Display



2.2.5 Remote TCP Jogging (option)

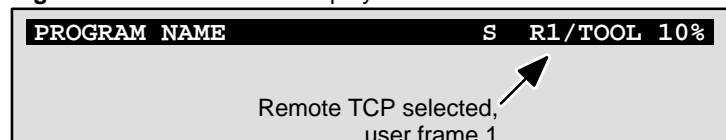
The optional remote TCP jog function allows you to control how the robot axes will jog. This is useful for applications in which the tool is fixed in the workcell and the robot manipulates the workpiece around the tool. The frame used for jogging is a user frame (UFRAME) you set up and select.

When remote TCP jogging is turned on, if you move the robot in x, y, or z using the TOOL coordinate system, the robot moves as it normally would without remote TCP jogging. If you move the robot in w, p, and r (rotational moves), the rotational center will be the remote TCP position.

In remote TCP jogging, an invisible tool is established to connect the faceplate to the remote TCP position. For example, if you select the WORLD coordinate system and jog the robot in w, the invisible tool will rotate along the WORLD x-axis. If you select the TOOL coordinate system and jog the robot in w, the invisible tool will rotate along the TOOL x-axis.

You select remote TCP jogging using the FCTN menu. You also select the remote TCP frame using the FCTN menu. When you select remote TCP jogging, “Rn/” appears next to the coordinate system name displayed on the teach pendant screen, where “n” is the number of the user frame, which can be from 1 to 5. Use Procedure 2-3 to select remote TCP jogging, select the remote TCP frame, and jog the axes.

Figure 2-8. Remote TCP Display



You can perform remote TCP jogging only if you have selected a Cartesian coordinate system such as WORLD, TOOL, JOGFRAME, or USER. You cannot perform remote TCP jogging if you have selected the JOINT coordinate system.

2.2.6 Motion Groups

A motion group defines different groups of axes that can be used for independent pieces of equipment, positioning tables, and other axes. There are three motion groups available. The controller can operate a maximum of 16 axes, however, only nine axes can belong to a single group. A robot is typically defined as Group 1.

When you create a program, you define the group mask which defines which groups the program will control. A program can be defined to use all three motion groups.

With multiple groups, the axes that jog depend on which group you have selected. You select groups using the FCTN menu or by pressing the SHIFT and COORD keys. Use Procedure 2-3 to select groups and jog the axes.

To change the group number, you can also use the jog menu. Refer to Section 2.2.8.

2.2.7 Extended Axes and Sub-Groups

Extended axes are the available axes controlled by the controller beyond the standard number of robot axes. There is a limit of three extended axes per motion group.

Extended axes become a sub-group of the motion group to which they belong. Because the teach pendant jog keys normally control the first six axes, you must first select the sub-group before you can jog an extended axis. For example, if the sub-group controls axes 7, 8, and 9, refer to Table 2-3 .

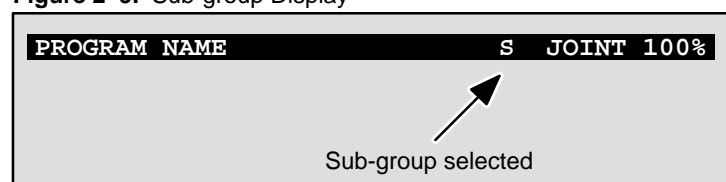
Table 2-3. Sub-Group Example

For Axis Number	Use Jog Keys
7	+X, -X
8	+Y, -Y
9	+Z, -Z

You select sub-groups using the FCTN menu. The status line at the top of the screen displays whether a sub-group is being used. See Figure 2-9. Use Procedure 2-3 to select sub-groups and jog the axes.

To change the sub-group number, you can also use the jog menu. Refer to Section 2.2.8.

Figure 2-9. Sub-group Display



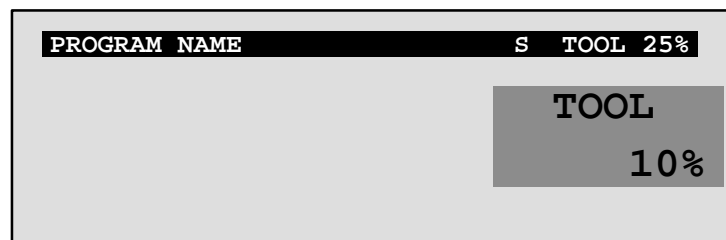
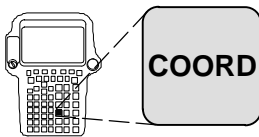
Use Procedure 2-3 to jog the robot and other axes.

Procedure 2-3 Jogging the Robot and Other Axes

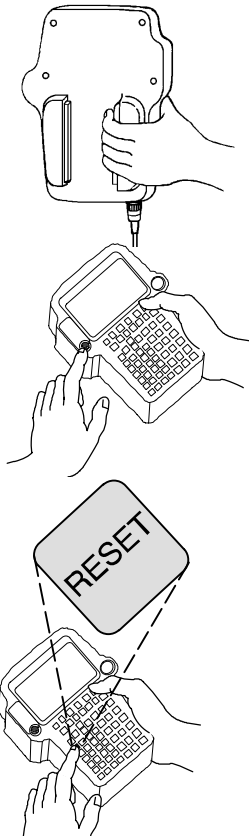
- Condition**
- All personnel and unnecessary equipment are out of the workcell.
 - All EMERGENCY STOP faults have been cleared. Refer to Section 7.1.1.
 - All other faults have been cleared and the fault light is not illuminated.

⚠ WARNING
Make certain that all safety requirements for your workplace have been followed; otherwise, damage to equipment or injury to personnel could occur.

- Step**
- 1 If your system is configured to have multiple groups, select the motion group that you want to jog.
 - 2 If your system is configured to have sub-groups, select the sub-group that you want to jog.
 - 3 **Select a coordinate system** by pressing the COORD key on the teach pendant until the coordinate system you want is displayed in the upper right hand corner of the teach pendant screen, and on the teach pendant LEDs. You will see a screen similar to the following.



NOTE The JOG SPEED value will automatically be set to 10%.



- 4 Hold the teach pendant and continuously press the DEADMAN switch on the back of the teach pendant.

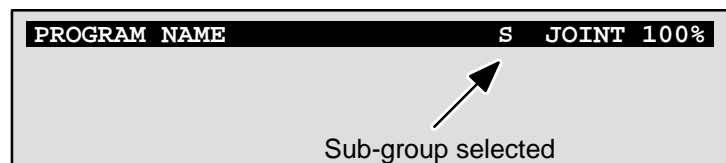
NOTE If you have the **Control Reliable (RS-1/RS-4) option** and you compress the DEADMAN switch fully, robot motion will not be allowed and an error occurs. This is the same as when the DEADMAN switch is released. **To clear the error**, press the DEADMAN switch in the center position and press RESET.

- 5 Turn the teach pendant ON/OFF switch to the ON position.

NOTE If you release the DEADMAN switch while the teach pendant is ON, an error will occur. **To clear the error**, continuously press the DEADMAN switch and then press the RESET key on the teach pendant.

- 6 **Change the group, select sub-group (extended axes), de-select sub-group (extended axes), or select wrist jog** by doing the following:

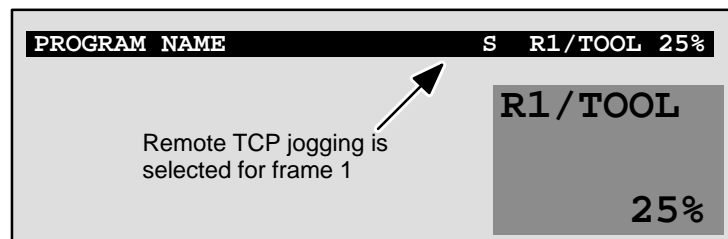
- a Press FCTN.
- b **To change the group**, move the cursor to CHANGE GROUP and press ENTER or press and hold the SHIFT key and press COORD.
- c **To select sub-group (extended axes)**, move the cursor to TOGGLE SUB-GROUP and press ENTER, or press the number of the function. See the following screen for an example.



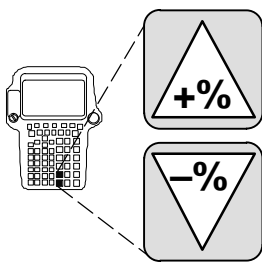
- d **To de-select sub-group (extended axes)**, move the cursor to TOGGLE SUB-GROUP and press ENTER, or press the number of the function.
- e **To select wrist jog**, move the cursor to TOGGLE WRIST JOG and press ENTER. The status line indicator for wrist jog is displayed in the upper right hand corner of the teach pendant screen.

7 If you want to use optional remote TCP jogging,

- a Select the Cartesian coordinate system you want to use for remote TCP jogging.
- b Press FCTN.
- c Move the cursor to TOGGLE REMOTE TCP and press ENTER. The status line indicator for remote TCP jogging is displayed as “Rn” in the upper right hand corner of the teach pendant screen next to the coordinate system, where “n” is the number of the remote TCP frame. See the following screen.



- d Press FCTN.
- e Move the cursor to CHANGE RTCP FRAME and press ENTER. Each time you select CHANGE RTCP FRAME, the user frame selection is advanced: from 1 (R1) to 2 (R2) to 3 (R3) to 4 (R4) to 5 (R5) and then back to 1 (R1). Select the user frame (UFRAME) you want to use for remote TCP jogging.
- f To de-select remote TCP jogging, press FCTN and move the cursor to TOGGLE REMOTE TCP and press ENTER.



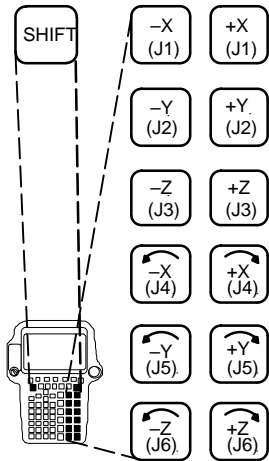
- 8 **Select a jog speed** by pressing and releasing the appropriate jog speed key until the jog speed you want is displayed in the upper right hand corner of the teach pendant screen. You will see a screen similar to the following.



NOTE Set the jog speed to a low percentage (%) value if you are inexperienced in jogging the robot, or if you are uncertain how the robot will move.

**WARNING**

In the next step, the robot will move. To stop the robot immediately any time during jogging, release the DEADMAN switch or press the EMERGENCY STOP button.



- 9 To jog, press and hold the SHIFT key and continuously press the jog key that corresponds to the direction in which you want to move the robot. To stop jogging release the jog key.
- 10 When you are finished jogging, turn the teach pendant ON/OFF switch to OFF, and release the DEADMAN switch.

2.2.8 Jog Menu

The *jog menu* provides a method to check and change the following jogging information:

- Currently selected frame number of each frame (TOOL, JOG, USER)
- Currently selected group number
- Currently selected sub-group type (ROBOT/EXT)

See Figure 2-10 for an illustration of the jog menu.

Figure 2-10. Jog Menu

TEST1		
UTILITY	TOOL	100%
	TOOL	2
	JOG	3
	USER	1
	GROUP	2
	ROBOT/EXT	

Use Procedure 2-4 to display and use the jog menu.

Procedure 2-4 Using the Jog Menu

- | | |
|-------------|---|
| Step | <ol style="list-style-type: none"> 1 To display the jog menu, press and hold the SHIFT key and press the COORD key. 2 Use the up and down arrow keys to move the cursor to the item you want to change. 3 To change the number of each frame, press the appropriate numeric key. The item on which the cursor is located is changed to the new value. Valid frame numbers are as follows: <ul style="list-style-type: none"> – TOOL, JOG 1-5 – USER 0-5 4 To change to sub-group (available only for systems with extended axes), move the cursor to ROBOT/EXT and press the left and right arrow keys. 5 To change the group number (available only for multiple motion group systems), move the cursor to GROUP and press the appropriate numeric key. You can specify numbers only for existing motion groups. 6 To close the jog menu, <ul style="list-style-type: none"> – Press SHIFT and COORD again. <p style="margin-left: 20px;">or</p> <ul style="list-style-type: none"> – Press the PREV key. |
|-------------|---|

You automatically close the jog menu after you enter a frame or group number.

3 GENERAL SETUP

Topics In This Chapter		Page
Inputs and Outputs (I/O) Setup	Inputs and Outputs (I/O) are electrical signals that enable the controller to communicate with the robot and external devices.	3-3
	<ul style="list-style-type: none"> • Analog I/O • Digital I/O • Group I/O • Robot I/O 	3-4 3-10 3-18 3-23
I/O Interconnect Setup	I/O Interconnect allows you to redirect robot digital inputs to system digital outputs or to redirect system digital inputs to robot digital outputs.	3-28
User Operator Panel (UOP) I/O Signals	User operator panel (UOP) signals allow you to connect most of the same signals as those on the standard operator panel to a remote operator panel or to a remote device.	3-33
	<ul style="list-style-type: none"> • UOP Input Signals • UOP Output Signals 	3-35 3-39
Model B I/O Setup	Model B I/O signals allow the controller to communicate with the robot and external devices.	3-43
	<ul style="list-style-type: none"> • Setting the DIP Switches • Setting Up the Basic Digital I/O Units • Setting Up User I/O • Digital I/O • Group I/O 	3-46 3-49 3-49 3-50 3-58
PLC I/O Setup	You can configure your system to allow the PLC to control the modular and fixed discrete I/O within the controller directly.	3-63
I/O Link Screen	You use the I/O Link screen to set up the Model B I/O unit and display the configuration of I/O link devices	3-70
	<ul style="list-style-type: none"> • I/O Link Device Screen • Model B I/O Detail Information • Setting Number of Ports 	3-70 3-72 3-74
Controlling I/O	Controlling I/O allows you to test the I/O in your system for proper function:	3-75
	<ul style="list-style-type: none"> • Forcing outputs • Simulating inputs and outputs 	3-75 3-76
Frames Setup	Frames can be set up to define positions in space relative to the robot, TCP, or workpiece. The frames that can be set are tool frame, user frame, and jog frame. World frame is predefined and cannot be changed.	3-78
	<ul style="list-style-type: none"> • Setting Up Tool Frame • Setting Up User Frame • Setting Up a Remote TCP Frame • Setting Up Jog Frame • Saving Frame Data 	3-80 3-94 3-110 3-120 3-129
Production Operation Setup	Production operation setup lets you set up programs to run during production.	3-131
	<ul style="list-style-type: none"> • Robot Service Request (RSR) • Program Number Select (PNS) 	3-131 3-134
Macro Commands	Macro commands allow you to write and set up a program by defining whether the program will execute from within another program, from the MANUAL FCTNS menu, from a teach pendant user key.	3-136
	<ul style="list-style-type: none"> • Setting Up Macro Commands • Executing Macro Commands 	3-136 3-141

Topics In This Chapter		Page
Axis Limits Setup	Axis limits change the robot default software travel limits.	3-145
Brake Timers Setup	Brake timers set the length of time the robot remains idle before the brakes are applied.	3-148
Brake On Hold Setup	Brake on hold enables or disables robot brake control when the robot is in a hold condition.	3-151
Current Language Setup	Current language allows you to change the current language. You can select from only those languages that have dictionaries.	3-152
Ignore Offset Setup	This function ignores the position offset specified by an OFFSET instruction. . .	3-153
Ignore Tool Offset Setup	This function ignores the position offset specified by a TOOL OFFSET instruction.	3-154
User Alarm Setup	User alarm setup allows you to define a message that will be displayed on the teach pendant status line when the UALM instruction is executed in a teach pendant program.	3-155
Override Select Setup	Override select setup allows you to specify four different clamps on the speed override of the robot.	3-158
Password Setup	Passwords (optional feature) prevent unauthorized personnel from changing critical aspects of the controller system. <ul style="list-style-type: none"> ● Install User Password Operations 3-163 ● Program and Setup User Password Operations 3-167 ● Password Log 3-170 ● Password Level Screen Permissions 3-173 	
Robot Payload Setting	You can set the payload of the robot, which is weight of the robot end-of-arm tooling. <ul style="list-style-type: none"> ● Payload Setting Process 3-176 ● Payload Setting Items 3-176 ● Payload Setup Procedure 3-178 ● Payload Teach Pendant Program Instruction 3-182 ● Inertia Equations 3-183 	

3.1 INPUTS AND OUTPUTS (I/O) SETUP

Inputs and Outputs (I/O) are electrical signals that enable the controller to communicate with the robot and external devices. There are seven kinds of user I/O:

- **Analog** – AI[n] and AO[n]
- **Digital** – DI[n] and DO[n]
- **Group** – GI[n] and GO[n]
- **Robot** - RI[n] and RO[n]
- **PLC** – PI[n] and PO[n]
- **SOP** – (can not be configured or addressed)
- **UOP** – UI[n] and UO[n]

These kinds of user I/O signals are attached to physical ports and are accessed from programs. The [n] corresponds to a signal number or group number. Setting up I/O establishes the correspondence between the signal number or group number and the physical port.

Robot inputs (RI) and outputs (RO) are preassigned. You cannot change the setup of RIs and ROs.

SOP inputs can be operated manually from the teach pendant and also monitored.

User I/O is available on Process I/O boards or Modular I/O. See Figure 3-1 and Figure 3-2. Process I/O boards contain multiple kinds of I/O such as analog I/O and digital I/O. Each module for Modular I/O can contain only a single type of user I/O.

Figure 3-1. Process I/O Board Hardware Layout

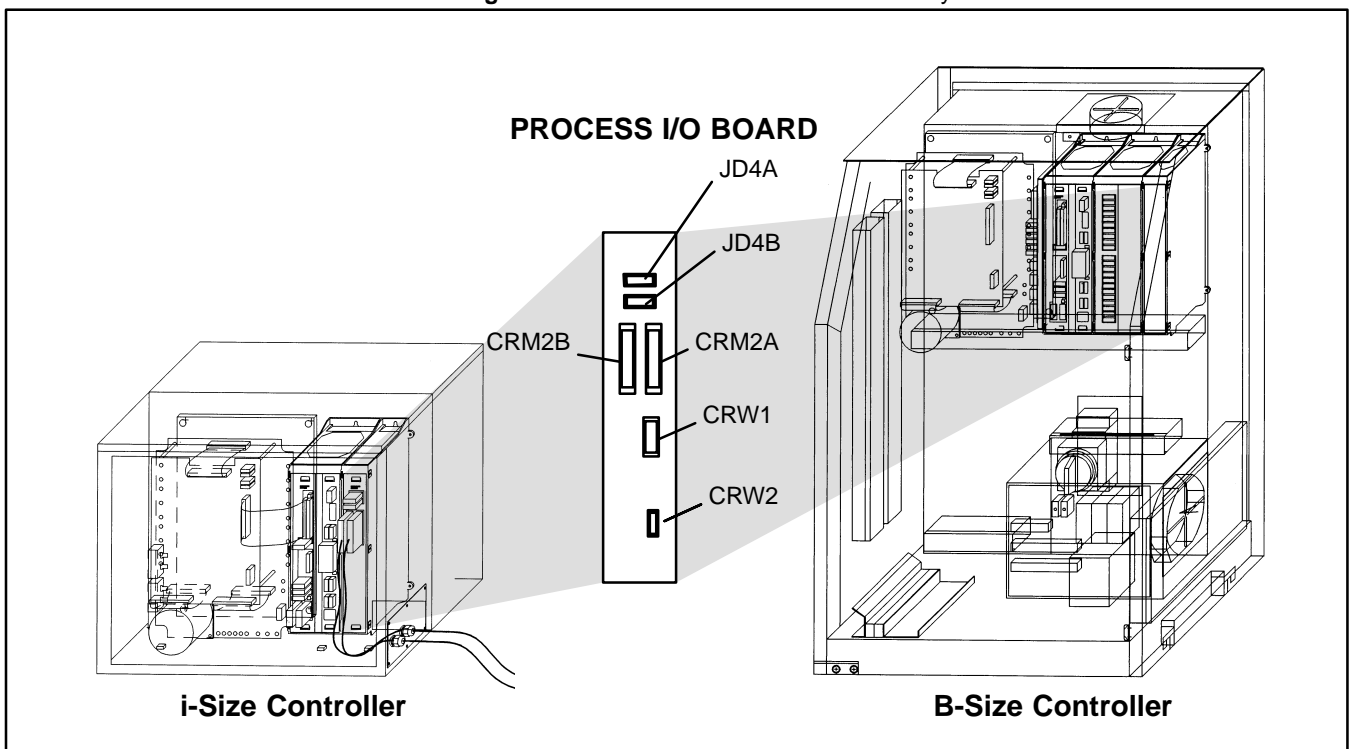
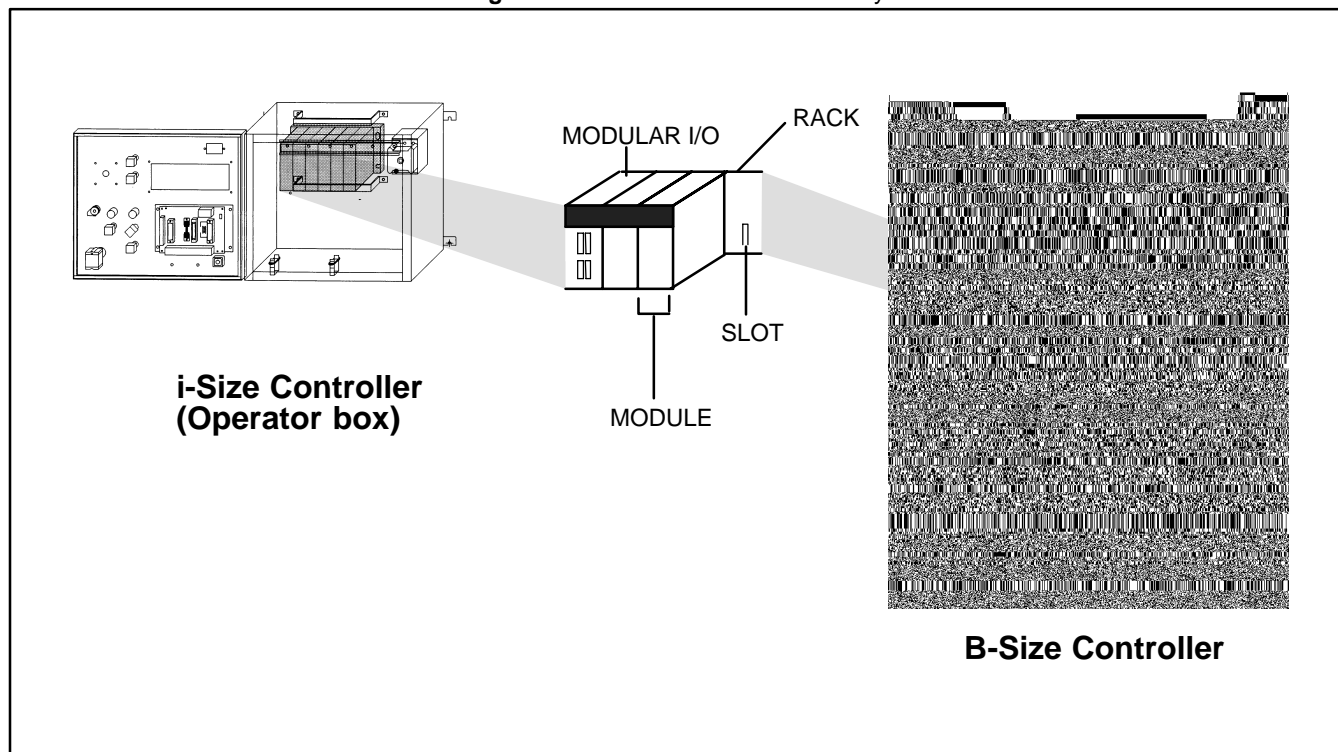


Figure 3-2. Modular I/O Hardware Layout



Robot Input (RI) and Robot Output (RO) signals are on the Main CPU board located on the backplane of the controller. These signals interface to the end effector through a cable that is plugged into the base of the robot and are accessed through the EE plug on the robot arm.

The GE Fanuc Genius I/O network and Allen Bradley Remote I/O are also available. For further information about these kinds of I/O refer to *A User's Guide to the FANUC Robotics SYSTEM R-J2 Controller Remote I/O Interface for an Allen-Bradley PLC*, *A User's Guide to the FANUC Robotics SYSTEM R-J2 Controller Genius Network Interface for GE Fanuc*, or to the *DeviceNet Interface Setup and Operations Manual*.

3.1.1 Analog I/O

An analog I/O signal is an input or output voltage that has a value within the range of the process I/O board or modular I/O, depending on the kind of I/O used.

You can:

- Configure analog I/O
- Control analog output
- Simulate analog I/O
- Add comments about analog I/O

Configuring Analog I/O

Each signal is configured to a rack, a slot in the rack, and the channel number when HandlingTool is loaded. See Figure 3-3 and Figure 3-4. You can change the configuration of

- **Rack** – the physical location on which the input or output process I/O board or modular I/O is mounted. Your system can contain multiple racks. Process I/O boards are always assigned Rack 0. Modular I/O begins at Rack 1.
- **Slot** – the space on the rack where the modular I/O module is connected. The slot number is also used to distinguish one process I/O board from another when more than one is used. The first process I/O board is always assigned as Slot 1.
- **Channel** – the physical position of the port on the process I/O board or modular I/O.

NOTE The GE Fanuc and Allen-Bradley I/O boards do not support Analog I/O.

Figure 3-3. Process I/O Board Hardware Layout for Analog I/O

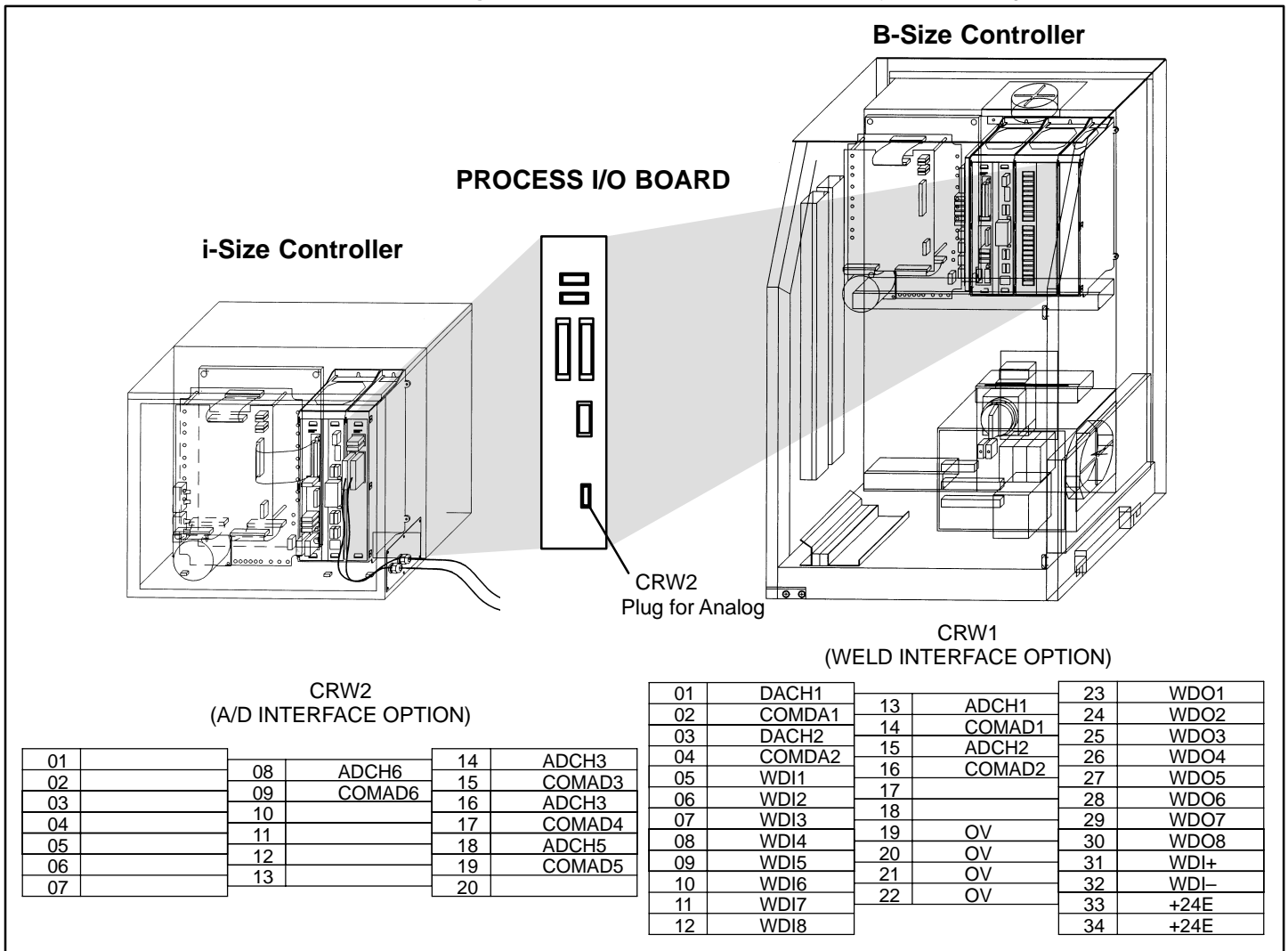
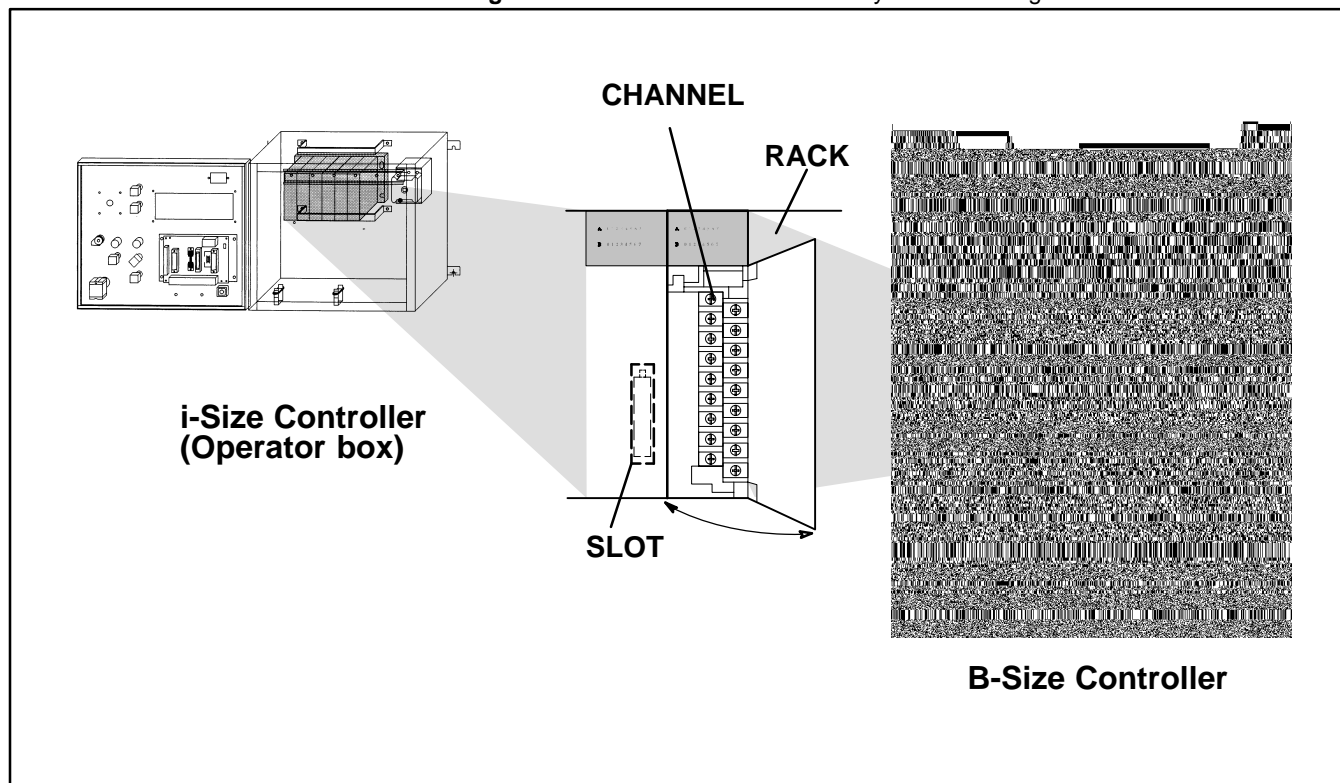


Figure 3-4. Modular I/O Hardware Layout For Analog I/O



Controlling Analog Outputs

Controlling outputs allows you to set the analog output value and turn it on in a program or to force it on manually. Analog I/O can be controlled individually. Refer to Chapter 6 to turn on output signals from a program, and Chapter 7 to force output signals.

Simulating Analog I/O

Simulating I/O allows you to test a program that uses I/O. Simulating I/O does not actually send output signals or receive input signals. Analog I/O signals can be simulated individually. Refer to Chapter 7.

Adding Comments About Analog I/O

Adding comments about analog I/O allows you to include text that describes the signal. For example, you can add a comment to indicate the line that is physically connected to the port.

Use Procedure 3-1 to configure analog I/O – rack, slot, and channel.

Procedure 3-1 Configuring Analog I/O – Rack, Slot, Channel

- Step**
- 1 Press MENUS.
 - 2 Select I/O.
 - 3 Press F1, [TYPE].
 - 4 Select Analog. You will see either the analog input or output screens. See the following screen for an example.

I/O Analog Out				JOINT	50 %
	#	SIM	VALUE		1/25
AO [1]	U	0	[]
AO [2]	U	0	[]
AO [3]	*	*	[]
AO [4]	*	*	[]
AO [5]	*	*	[]
AO [6]	*	*	[]
AO [7]	*	*	[]
AO [8]	*	*	[]
AO [9]	*	*	[]
AO [10]	*	*	[]
[TYPE]	CONFIG	IN/OUT	SIMULATE	UNSIM	

To change between the display of the input and output screens, press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

- 5 Move the cursor to the I/O signal you want to configure.
- 6 Press F2, CONFIG. You will see a screen similar to the following.

I/O Analog Out				JOINT	50 %
AO #	RACK	SLOT	CHANNEL		1/25
1	0	1	1		
2	0	1	2		
3	0	0	0		
4	0	0	0		
5	0	0	0		
6	0	0	0		
7	0	0	0		
8	0	0	0		
9	0	0	0		
[TYPE]	MONITOR	IN/OUT	DETAIL	HELP	>
[TYPE]	VERIFY				>

7 Configure the I/O:

- a** Move the cursor to RACK, type the value, and press ENTER.
- b** Move the cursor to SLOT, type the value, and press ENTER.
- c** Move the cursor to CHANNEL, type the value, and press ENTER.

8 To add a comment:

- a** Press F4, DETAIL. You will see a screen similar to the following.

I/O Analog Out		JOINT	50 %
Analog Output Detail		1/4	
Analog Output: AO [1]			
1	Rack Number:	0	
2	Slot Number:	1	
3	Channel:	1	
4	Comment:	[]
[TYPE]	NEXT	IN/OUT	>
[TYPE]	VERIFY		>

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

9 To determine if the assignment is valid, press NEXT, >, and then press F2, VERIFY.

- If the assignment is valid, the message, "Port assignment is valid," is displayed.
- If the assignment is not valid, the message, "Port assignment is invalid," is displayed.

 **CAUTION**

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving I/O Information

- 10** To save the information (when all I/O is configured):
 - a** Press **MENUS**.
 - b** Select **FILE**.
 - c** Press **F1**, [**TYPE**].
 - d** Select **File**.
 - e** Press **F5**, [**UTIL**].
 - f** Select **Set Device**.
 - g** Move the cursor to the device you want and press **ENTER**.
 - h** Press **FCTN**.
 - i** Select **SAVE**. The file will be saved to the **DIOCFGSV.IO** file on the default device.

 **WARNING**

You must turn off the controller and then turn it back on to use the new information; otherwise, you could injure personnel or damage equipment.

- 11** When you are finished configuring I/O, turn off the controller. Turn on the controller so it can use the new information.

3.1.2 Digital I/O

Digital I/O signals provide access to data on a single input or output signal line. Digital I/O signals can be ON or OFF.

NOTE If you configure UOP input and output signals, the UI and UO physical locations are actually **digital** I/O on the process I/O board or modular I/O. In effect, the physical digital I/O can be double configured as both user I/O and digital I/O logicals. This allows you to control or monitor user I/O signals within your program using the digital I/O instructions.

You can:

- Configure digital I/O
- Simulate digital I/O
- Control digital outputs
- Add comments about digital I/O

Complementary Output Signals

You can configure digital output signals to be controlled independently or in complementary pairs. If an output signal is controlled independently, a command to turn that output signal on or off controls only that output signal. If an output signal is controlled in a complementary pair, a command to turn that signal ON will also turn its pair OFF. A command to turn the signal OFF will also turn its pair ON.

Polarity

You can configure digital input/output signals with normal polarity (active ON) or inverse polarity (active OFF).

Configuring Digital I/O

Each signal is configured to a rack, a slot in the rack, and the starting point for numbering when HandlingTool is loaded. Digital I/O is configured in groups of eight. See Figure 3-5 and Figure 3-6. You can change the configuration of the

- **Rack** – the physical location on which the input or output process I/O board or modular I/O is mounted. Your system can contain multiple racks. Process I/O boards are always assigned Rack 0. Modular I/O begins at Rack 1. The I/O assigned to the Allen-Bradley or GEFANUC remote I/O board, is configured as Rack 16. The I/O assigned to the DeviceNet Interface is configured as Racks 81 – 84.
- **Slot** – the space on the rack where the modular I/O module is connected. The slot number is also used to distinguish one process I/O board from another when more than one is used. The first process I/O board is always assigned as Slot 1. The I/O assigned to the Allen-Bradley or GEFANUC remote I/O board, is assigned as Slot 1. The slot number for DeviceNet Interface I/O is the MAC Id for the device.
- **Starting Point** – the physical position on the process I/O board or modular I/O board of the first port in a range of input or output signals. Valid starting points are 1, 9, 17, 25 and so forth.

Figure 3-5. Process I/O Board Hardware Layout for Digital I/O

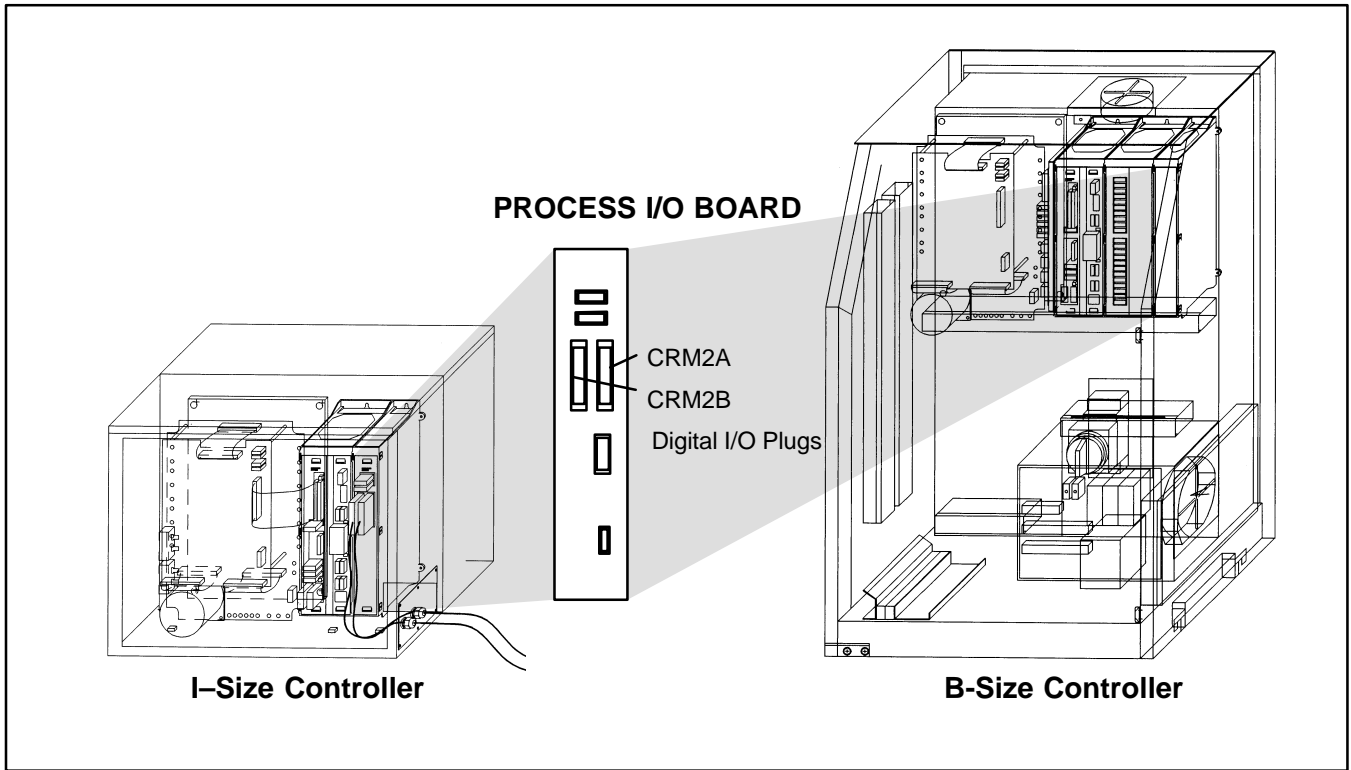
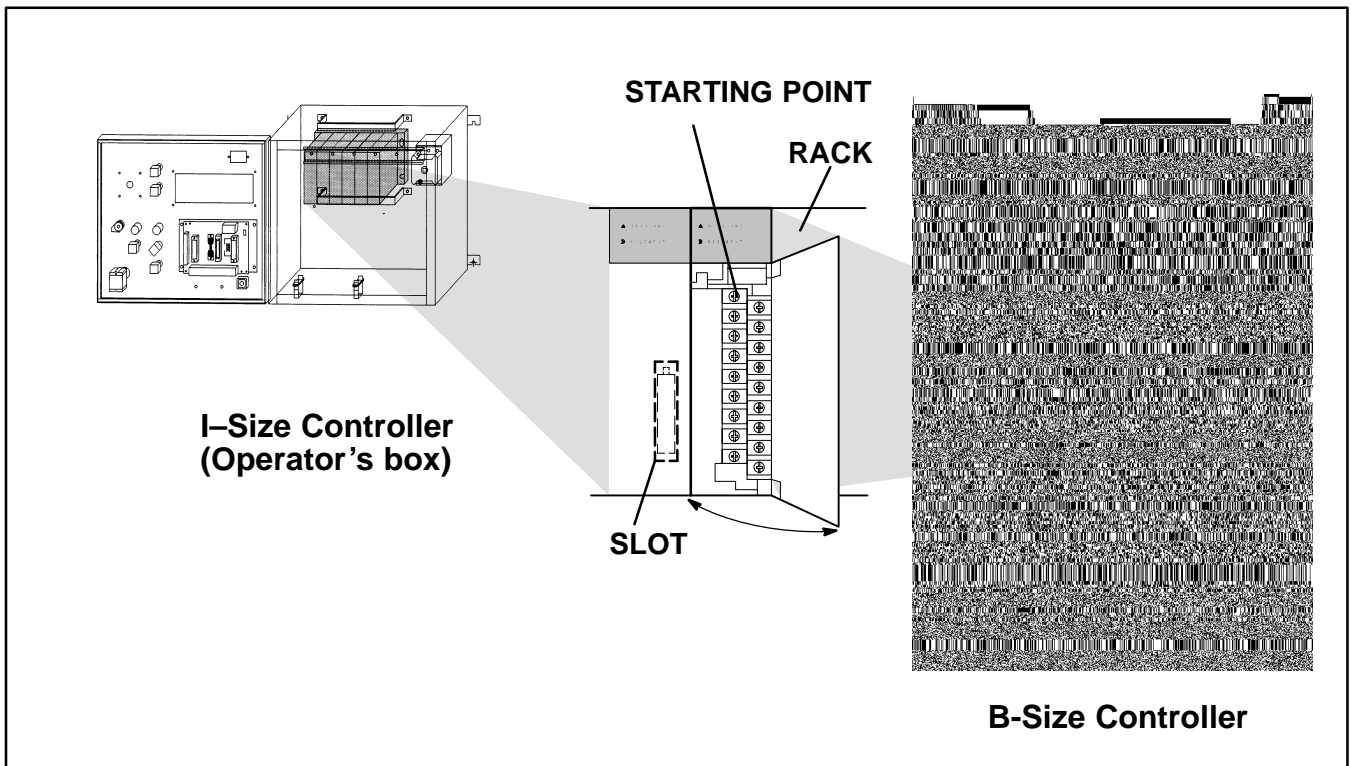


Figure 3-6. Modular I/O Hardware Layout For Digital I/O



Controlling Digital Outputs

Controlling outputs allows you to set the digital output value and turn it on in a program or to force it on manually. Digital I/O can be controlled individually. Refer to Chapter 6 to turn on output signals from a program, and Chapter 7 to force output signals.

Simulating Digital I/O

Simulating I/O allows you to test a program that uses I/O. Simulating I/O does not actually send output signals or receive input signals. Digital I/O can be simulated individually. Refer to Chapter 7.

Adding Comments About Digital I/O

Adding comments about digital I/O allows you to include text that describes the signal. For example, you can add a comment to indicate the line that is physically connected to the port.

Use Procedure 3-2 to configure digital I/O – rack, slot and start point.
Use Procedure 3-3 to configure digital I/O – polarity and complementary pairs.

Procedure 3-2 Configuring Digital I/O – Rack, Slot, Start Point

NOTE Digital I/O is configured by the system. Use this procedure if you want to change the configuration.

- Step**
- 1 Press MENUS.
 - 2 Select I/O.
 - 3 Press F1, [TYPE].
 - 4 Select Digital. You will see either the digital input or digital output screens. See the following screen for an example.

I/O Digital In				JOINT	50 %
	#	SIM	STATUS	1/256	
DI [1]	*	OFF []	
DI [2]	*	OFF []	
DI [3]	*	OFF []	
DI [4]	*	OFF []	
DI [5]	*	OFF []	
DI [6]	*	OFF []	
DI [7]	*	OFF []	
DI [8]	*	OFF []	
DI [9]	*	OFF []	
DI [10]	*	OFF []	
[TYPE] CONFIG IN/OUT SIMULATE UNSIM					

To change between the display of the input and output screens, press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

- 5 Press F2, CONFIG. You will see a screen similar to the following.

I/O Digital In				JOINT	50 %
#	RANGE	RACK	SLOT	1/32	
				START PT	
1	DI [1 - 8]	0	1	21	
2	DI [9 - 16]	0	1	29	
3	DI [17 - 24]	0	1	37	
4	DI [25 - 32]	0	0	0	
5	DI [33 - 40]	0	0	0	
6	DI [41 - 48]	0	0	0	
7	DI [49 - 56]	0	0	0	
8	DI [57 - 64]	0	0	0	
9	DI [65 - 72]	0	0	0	
[TYPE] MONITOR IN/OUT DETAIL HELP >					
[TYPE] VERIFY >					

6 Configure the I/O:

- a** Move the cursor to RACK, type the value, and press ENTER.
- b** Move the cursor to SLOT, type the value, and press ENTER.
- c** Move the cursor to START PT, type the value, and press ENTER.

7 To add a comment:

- a** Press F4, DETAIL. You will see a screen similar to the following.

JOINT 50 %	
Digital Input Detail 1/19	
Digital Input: DI [1]	
Digital inputs: [1 - 8]	
1	Rack number:
2	Slot number: 1
3	Starting point: 0
4	Comment: [1] []
5	Comment: [2] []
6	Comment: [3] []
[TYPE]	NEXT IN/OUT >
[TYPE]	VERIFY >

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

8 To determine if the assignment is valid, press NEXT, >, and then press F2, VERIFY.

- If the assignment is valid, the message, "Port assignment is valid," is displayed.
- If the assignment is not valid, the message, "Port assignment is invalid," is displayed.

⚠ CAUTION

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving I/O Information

9 To save the information (when all I/O is configured):

NOTE Make sure the digital I/O menu is displayed.

- a** Press **MENUS**.
- b** Press **F1**, [**TYPE**].
- c** Select **File**.
- d** Press **F5**, [**UTIL**].
- e** Select **Set Device**.
- f** Move the cursor to the device you want and press **ENTER**.
- g** Press **FCTN**.
- h** Select **SAVE**. The file will be saved to the **DIOCFGSV.IO** file on the default device.

⚠ WARNING

You must turn off the controller and then turn it back on to use the new information; otherwise, you could injure personnel or damage equipment.

- 10** When you are finished configuring I/O, turn off the controller. Turn on the controller so it can use the new information.

Procedure 3-3 Configuring Digital I/O – Polarity and Complementary Pairs

- Step**
- 1 Press **MENUS**.
 - 2 Select **I/O**.
 - 3 Press **F1**, [**TYPE**].
 - 4 Select **Digital**. You will see either the digital input or digital output screen. See the following screen for an example.

I/O Digital In				JOINT	50 %
#	SIM	STATUS	1/256		
DI [1]	*	OFF []		
DI [2]	*	OFF []		
DI [3]	*	OFF []		
DI [4]	*	OFF []		
DI [5]	*	OFF []		
DI [6]	*	OFF []		
DI [7]	*	OFF []		
DI [8]	*	OFF []		
DI [9]	*	OFF []		
DI [10]	*	OFF []		
[TYPE]	CONFIG	IN/OUT	SIMULATE	UNSIM	

To change the display between the Digital Input and Digital Output screen press **F3**, **IN/OUT**.

To move quickly through the information, press and hold the **SHIFT** key and press the down or up arrow keys.

- 5 Press **F2**, **CONFIG**. You will see a screen similar to the following.

I/O Digital In				JOINT	50 %
#	RANGE	RACK	SLOT	1/32	
				START	PT
1	DI [1 - 8]	1	1	1	
2	DI [9 - 16]	1	1	9	
3	DI [17 - 24]	1	1	17	
4	DI [25 - 32]	1	1	25	
5	DI [33 - 40]	1	1	33	
6	DI [41 - 48]	*	*	*	
7	DI [49 - 56]	*	*	*	
8	DI [57 - 64]	*	*	*	
9	DI [65 - 72]	*	*	*	
[TYPE]	MONITOR	IN/OUT	DETAIL	HELP >	
[TYPE]	VERIFY	>			

6 Move the cursor to the input or output group you want to configure.

7 Press F4, DETAIL.

8 To set polarity,

```
12 Polarity:[1] [ INVERSE ]
```

a Move the cursor to the polarity of the signal you want to set. You will see a screen similar to the following.

JOINT 50 %			
Digital	Output	Detail	12/23
12	Polarity:	[1] INVERSE	
13	Polarity:	[2] NORMAL	
14	Polarity:	[3] NORMAL	
15	Polarity:	[4] NORMAL	
16	Polarity:	[5] NORMAL	
17	Polarity:	[6] NORMAL	
18	Polarity:	[7] NORMAL	
19	Polarity:	[8] NORMAL	
20	Complementary:	[1 - 2] TRUE	
21	Complementary:	[3 - 4] TRUE	
[TYPE] NEXT IN/OUT INVERSE NORMAL >			
[TYPE] VERIFY INVERSE NORMAL >			

b Select the polarity you want:

- For inverse polarity, press F4, INVERSE.
- For normal polarity, press F5, NORMAL.

9 To set complementary pairs (digital output signals only),

```
20 Complementary:[1-2] FALSE ]
```

a Move the cursor to the pair you want to set. You will see a screen similar to the following.

JOINT 50 %			
Digital	Output	Detail	20/23
14	Polarity:	[3] NORMAL	
15	Polarity:	[4] NORMAL	
16	Polarity:	[5] NORMAL	
17	Polarity:	[6] NORMAL	
18	Polarity:	[7] NORMAL	
19	Polarity:	[8] NORMAL	
20	Complementary	[1 - 2] FALSE	
21	Complementary	[3 - 4] TRUE	
22	Complementary	[5 - 6] TRUE	
23	Complementary	[7 - 8] TRUE	
[TYPE] NEXT IN/OUT TRUE FALSE >			
[TYPE] VERIFY TRUE FALSE >			

b Select the complementary value:

- For no complementary pair, press F5, FALSE.
- For a complementary pair, press F4, TRUE.

⚠ CAUTION

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving I/O Information 10 To save the information (when all I/O is configured):

NOTE Make sure the digital I/O menu is displayed.

- a Press **MENUS**.
- b Select **FILE**.
- c Press **F1**, [**TYPE**].
- d Select **File**.
- e Press **F5**, [**UTIL**].
- f Select **Set Device**.
- g Move the cursor to the device you want and press **ENTER**.
- h Press **FCTN**.
- i Select **SAVE**. The file will be saved to the **DIOCFGSV.IO** file on the default device.

⚠ WARNING

You must turn off the controller and then turn it back on to use the new information; otherwise, you could injure personnel or damage equipment.

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

3.1.3 Group I/O

Group I/O signals provide access to data on more than one input or output signal line at one time. Group I/O instructions allow a program to monitor or set a group of input or output signals as a binary number.

NOTE If you configure UOP input and output signals, the user I/O physical locations are actually digital I/O on the process I/O board or modular I/O. In effect, the physical digital I/O can be double configured as both user I/O and digital I/O logicals. If you then group your digital I/O signals, you can control or monitor user I/O signals within your program using the group I/O instructions. For example, you can configure the UOP signals into groups and issue a single command to control the entire group.

If you want to use group I/O, you must configure group I/O. You can also:

- Control group outputs
- Simulate group I/O
- Add comments about group I/O

Configuring Group I/O

Each group must be configured to a rack, a slot in the rack, the starting point for numbering, and the number of points when HandlingTool is loaded. See Figure 3-7 and Figure 3-8. You can change the configuration of:

- **Rack** – the physical location on which the input or output process I/O board or modular I/O is mounted. Your system can contain multiple racks. Process I/O boards are always assigned Rack 0. Modular I/O begins at Rack 1.
- **Slot** – the space on the rack where the modular I/O module is connected. The slot number is also used to distinguish one process I/O board from another when more than one is used.
- **Starting Point** – the physical position on the process I/O board or modular I/O of the first port to be included in the group. The starting point can be any number up to and including 999.
- **Number of Points** – indicates how many inputs or outputs will be in a group. The lowest number, or starting point, of the input or output is the least significant bit. The number of points can be from 1 up to and including 16.

Figure 3-7. Process I/O Board Hardware Layout for Group I/O

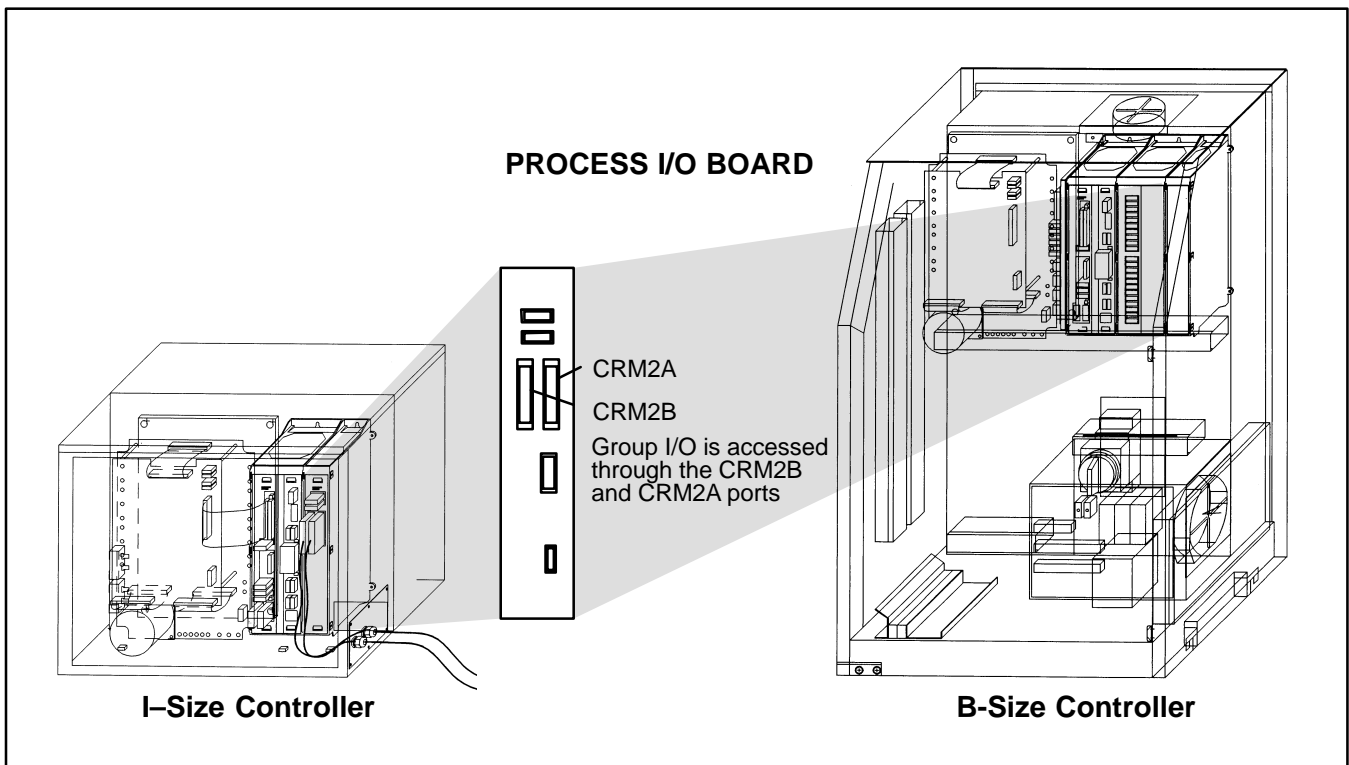
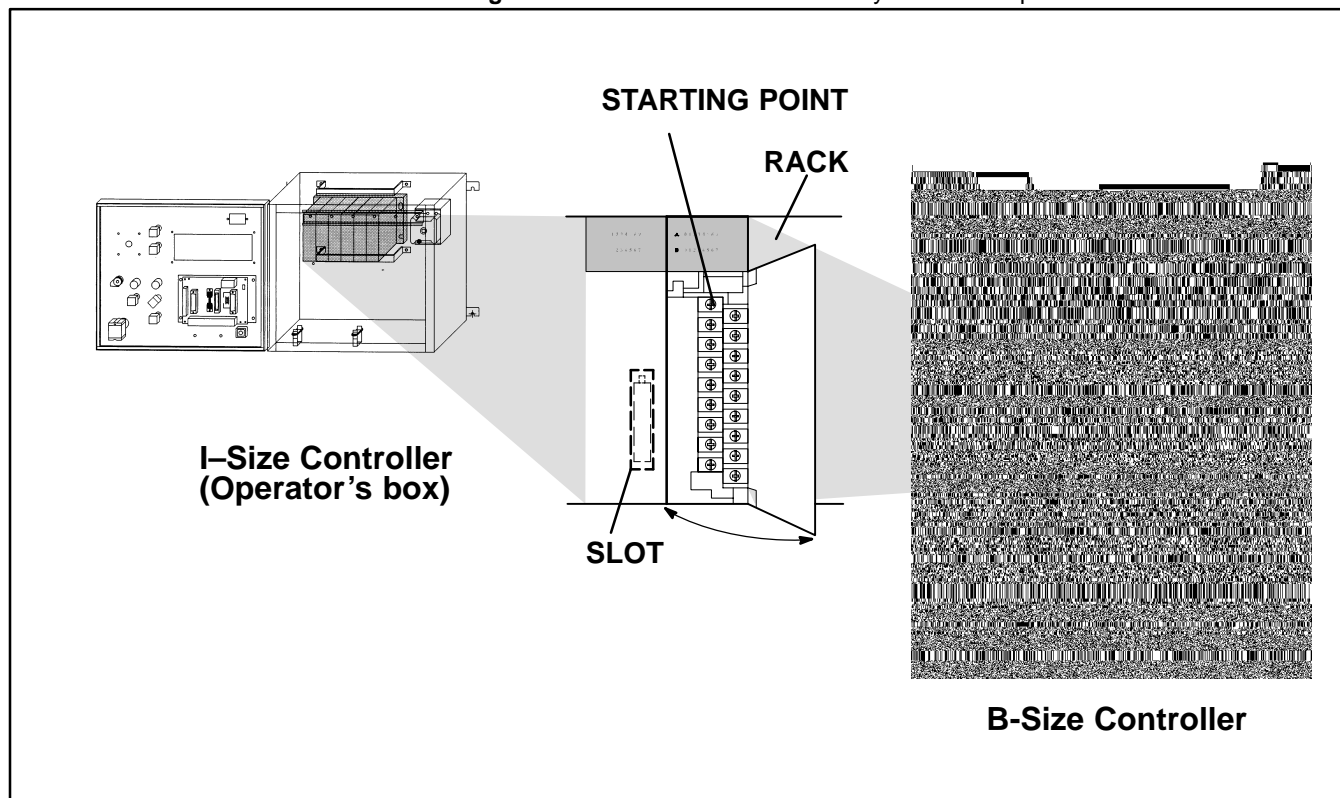


Figure 3-8. Modular I/O Hardware Layout For Group I/O



Controlling Group Outputs

Controlling outputs allows you to set the group output value and turn it on in a program or to force it on manually. Refer to Chapter 6 to turn on output signals from a program, and Chapter 7 to force output signals.

Simulating Group I/O

Simulating I/O allows you to test a program that uses I/O. Simulating I/O does not actually send output signals or receive input signals. Refer to Chapter 7.

Adding Comments About Group I/O

Adding comments about group I/O allows you to include text that describes the signal. For example, you can add a comment to indicate why you are grouping the signals.

Use Procedure 3-4 to configure group I/O – rack, slot, start point, and number of points.

Procedure 3-4 Configuring Group I/O – Rack, Slot, Start Point, Num Pts

- Step**
- 1 Press MENUS.
 - 2 Select I/O.
 - 3 Press F1, [TYPE].
 - 4 Select Group. You will see either the group input or group output screens. See the following screen for an example.

I/O Group Out			JOINT	50 %
#	SIM	VALUE		1/25
GO [1]	*	0 []
GO [2]	*	0 []
GO [3]	*	0 []
GO [4]	*	0 []
GO [5]	*	0 []
GO [6]	*	0 []
GO [7]	*	0 []
GO [8]	*	0 []
GO [9]	*	0 []
GO [10]	*	0 []

[TYPE] CONFIG IN/OUT SIMULATE UNSIM

To change between the display of the input and output screens, press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

- 5 Press F2, CONFIG. You will see a screen similar to the following.

I/O Group Out				JOINT	50 %
GO #	RACK	SLOT	START PT	NUM	PTS
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0

[TYPE] MONITOR IN/OUT DETAIL HELP >

[TYPE] VERIFY >

6 Configure the I/O:

- a** Move the cursor to RACK, type the value, and press ENTER.
- b** Move the cursor to SLOT, type the value, and press ENTER.
- c** Move the cursor to START PT, type the value, and press ENTER.
The starting point can be any number up to and including 999.
- d** Move the cursor to NUM PTS, type the value, and press ENTER.
The number of points can be from 1 up to and including 16.

7 To add a comment:

- a** Press F4, DETAIL. You will see a screen similar to the following.

I/O	Group	Out	JOINT	50	%
Group Output Detail			1/5		
Group Output: GO [1]					
1	Rack Number:		0		
2	Slot Number:		0		
3	Starting Point:		0		
4	Number of Points:		0		
5	Comment:		[]	
[TYPE]	NEXT	IN/OUT	>		
[TYPE]	VERIFY	>			

- b** Move the cursor to the comment line and press the ENTER key.
 - c** Select a method of naming the comment.
 - d** Press the appropriate function keys to add the comment.
 - e** When you are finished , press ENTER.
- 8** To determine if the assignment is valid, press NEXT, >, and then press F2, VERIFY.
- If the assignment is valid, the message, “Port assignment is valid,” is displayed.
 - If the assignment is not valid, the message, “Port assignment is invalid,” is displayed.

⚠ CAUTION

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving I/O Information

9 To save the information (when all I/O is configured):

NOTE Make sure the group I/O menu is displayed.

- a** Press **MENU**.
- b** Select **FILE**.
- c** Press **F1**, [**TYPE**].
- d** Select **File**.
- e** Press **F5**, [**UTIL**].
- f** Select **Set Device**.
- g** Move the cursor to the device you want and press **ENTER**.
- h** Press **FCTN**.
- i** Select **SAVE**. The file will be saved to the **DIOCFGSV.IO** file on the default device.

⚠ WARNING

You must turn off the controller and then turn it back on to use the new information; otherwise, you could injure personnel or damage equipment.

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

3.1.4 Robot I/O

The robot I/O screen indicates the status of the robot I/O. Robot I/O consists of the input and output signals between the controller and the robot. These signals are sent through the EE (End Effector) connector located on the robot. You can change the status of outputs at this screen.

You can configure

- Complementary RO signals
- Polarity of RI/RO signals

Complementary Output Signals

You can configure robot output signals to be controlled independently or in complementary pairs. If an output signal is controlled independently, a command to turn that output signal on or off controls only that output signal. If an output signal is controlled in a complementary pair, a command to turn that signal on will also turn its pair off. A command to turn the signal off will also turn its pair on.

Polarity

You can configure robot input/output signals with normal polarity (active ON) or inverse polarity (active OFF). Use Procedure 3-5 to configure robot I/O.

Procedure 3-5 Configuring Robot I/O

- Step**
- 1 Press MENUS.
 - 2 Select I/O.
 - 3 Press F1, [TYPE].
 - 4 Select Robot. You will see either the robot input or robot output screens. See the following screen for an example.

I/O	Robot	Out	JOINT	10 %
	#	STATUS		1/24
RO[1]	OFF	[]
RO[2]	OFF	[]
RO[3]	OFF	[]
RO[4]	OFF	[]
RO[5]	OFF	[]
RO[6]	OFF	[]
RO[7]	OFF	[]
RO[8]	OFF	[]
RO[9]	*	[]
RO[10]	*	[]
[TYPE]	DETAIL	IN/OUT ON OFF

To change between the display of the input and output screens, press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

- 5 **To force an output signal, move the cursor to the output you want to change and press**
 - F4, ON, to turn on an output signal.
 - F5, OFF, to turn off an output signal.

6 Press F2, DETAIL. You will see a screen similar to the following.

I/O	Robot Out		JOINT	50 %
Robot Output Detail			29/60	
21	Comment:	[21]	[]
22	Comment:	[22]	[]
23	Comment:	[23]	[]
24	Comment:	[24]	[]
25	Polarity:	[1]	NORMAL	
26	Polarity:	[2]	NORMAL	
27	Polarity:	[3]	NORMAL	
28	Polarity:	[4]	NORMAL	
29	Polarity:	[5]	NORMAL	
[TYPE] MONITOR IN/OUT INVERSE NORMAL				

NOTE In the robot I/O detail screens,
Items 1–24 are comments.
Items 25–48 are polarity.
Items 49–60 are complementary pairs.

7 To add a comment,

- a** Move the cursor to the comment line that corresponds to the robot signal number you want and press the ENTER key.
- 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.

8 To set polarity,9 Polarity:[1] [**INVERSE**]

- a** Move the cursor to the polarity of the signal you want to set. See the following screen for an example.

I/O	Robot Out		JOINT	50 %
Robot Output Detail			25/60	
25	Polarity:	[1]	INVERSE	
26	Polarity:	[2]	NORMAL	
27	Polarity:	[3]	NORMAL	
28	Polarity:	[4]	NORMAL	
29	Polarity:	[5]	NORMAL	
30	Polarity:	[6]	NORMAL	
31	Polarity:	[7]	NORMAL	
32	Polarity:	[8]	NORMAL	
33	Polarity:	[9]	NORMAL	
[TYPE] MONITOR IN/OUT INVERSE NORMAL				

- b** Select the polarity you want:

- For inverse polarity, press F4, INVERSE.
- For normal polarity, press F5, NORMAL.

9 To set complementary pairs (robot output signals only),27 Complementary:[1-2] [**FALSE**]

- a** Move the cursor to the complementary pair you want to set. See the following screen for an example.

I/O	Robot Out		JOINT	50 %
Robot Output Detail			49/60	
43	Polarity:	[19]	NORMAL	
44	Polarity:	[20]	NORMAL	
45	Polarity:	[21]	NORMAL	
46	Polarity:	[22]	NORMAL	
47	Polarity:	[23]	NORMAL	
48	Polarity:	[24]	NORMAL	
49	Complementary	[1-2]	FALSE	
50	Complementary	[3-4]	TRUE	
51	Complementary	[5-6]	TRUE	
52	Complementary	[7-8]	TRUE	
[TYPE] MONITOR IN/OUT TRUE FALSE >				

NOTE The number of RO signals varies depending on the kind of robot.

- b** Select the complementary value:

- For no complementary pair, press F5, FALSE.
- For a complementary pair, press F4, TRUE.

⚠ CAUTION

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving I/O Information

10 To save the information (when all I/O is configured):

NOTE Make sure the robot I/O menu is displayed.

- a** Press **MENUS**.
- b** Select **FILE**.
- c** Press **F1**, [**TYPE**].
- d** Select **File**.
- e** Press **F5**, [**UTIL**].
- f** Select **Set Device**.
- g** Move the cursor to the device you want and press **ENTER**.
- h** Press **FCTN**.
- i** Select **SAVE**. The file will be saved to the **DIOCFGSV.IO** file on the default device.

⚠ WARNING

You must turn off the controller and then turn it back on to use the new information; otherwise, you could injure personnel or damage equipment.

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

3.2 I/O INTERCONNECT SETUP

The optional I/O interconnect feature allows you to output the states of robot digital input (RI) and digital input (DI) signals to digital output (DO) and robot output (RO) signals to notify external devices of the input states of the signals.

With I/O InterConnect, you can do the following:

- Redirect the status of a RI signal to a DO signal
RI[m] -> DO[n], where
m: RI signal number
n: 0-999
- Redirect the status of a DI signal to a RO signal
DI[i] -> RO[j], where
i: 0-999
j: RO signal number
- Redirect the status of a DI signal to a DO signal
DI[k] -> DO[l], where
k: 0-999
l: 0-999
- Redirect the status of an SI signal to a DO signal
SI[q] -> DO[r], where
q: SI signal number
r: 0-999
- Redirect the status of an emergency stop (ES) signal to a DO signal
ES -> DO[t], where
ES: emergency stop signal
t: 0-999

You use the I/O InterConnect screen to connect signals and enable and disable the connections.

For example, when “ENABLE DI[2]->RO[3]” is set, the state of DI[2] is output to RO[3].

NOTE I/O interconnection changes take effect immediately. It is **NOT** necessary to turn the controller on then off for these changes to take effect.

Restrictions

You have the following restrictions when you use I/O InterConnect:

- When the redirection of DI[i] to DO[j] is enabled, the state of DI[i] is periodically output to DO[j]. In this case, DO[j] cannot be changed from the teach pendant or by a program.
- The redirection of each signal can be enabled or disabled only from the relevant page of the I/O Interconnect screen.
- If two or more input signals are redirected to an output signal, the state of each input signal is redirected to the output signal.

```
1 ENABLE      RI[ 1 ] -> DO[ 1 ]
2 ENABLE      RI[ 2 ] -> DO[ 1 ]
```

For example, if the signals are redirected as above, the output of DO[1] will be unpredictable when RI[1] is turned on and RI[2] is turned off (actually, DO[1] is repeatedly turned on and off).

Table 3-1 lists and describes each item on the I/O InterConnect screen. The I/O InterConnect screens are shown in Procedure 3-6 .

Table 3-1. I/O Interconnect Screen Items

ITEM	DESCRIPTION
No.	Line number of the interconnect. The ITEM key can be used to select a particular line.
Enb/Disabl	Specifies whether to redirect the signal. If set to ENABLE the signal will be redirected. If set to DISABLED , the signal will not be redirected. If the signal number of the DO or the DI is 0, then the signal will not be redirected .
Input	Displays the RI or DI signal that will be redirected. RI signals cannot be modified.
Output	Displays the RO or DO signal that will receive the status for the input signal. RO signals cannot be modified.

Use Procedure 3-6 to use interconnect I/O.

Procedure 3-6 Using Interconnect I/O

- Step**
- 1 Press MENUS.
 - 2 Select I/O.
 - 3 Press F1, [TYPE].
 - 4 Select Inter Conect. You will see a screen similar to the following.

INTER CONNECT		JOINT 100%	
		1/24	
No.	Enb/Disabl	INPUT	OUTPUT
1	ENABLE	RI [1] ->	DO [0]
2	DISABLE	RI [2] ->	DO [0]
3	DISABLE	RI [3] ->	DO [0]
4	DISABLE	RI [4] ->	DO [0]
5	DISABLE	RI [5] ->	DO [0]
6	DISABLE	RI [6] ->	DO [0]
24	DISABLE	RI [24] ->	DO [0]
[TYPE]		[SELECT]ENABLE DISABLE	

- RI → DO** **5** Press F3, [SELECT]. If RI → DO had been selected previously, you will see a screen similar to the following:

INTER CONNECT		JOINT 100%										
			1/24									
No.	Enb/Disabl	INPUT	OUTPUT									
1	DISABLE	RI [1] ->	DO [0]									
2	DISABLE	RI [2] ->	DO [0]									
3	DISABLE	RI [3] ->	DO [0]									
4	DISABLE	RI [4] ->	DO [0]									
5	DISABLE	RI [5] ->	DO [0]									
6	DISABLE	RI [6] ->	DO [0]									
24	DISABLE	<table border="1" style="margin-left: 20px;"> <tr> <td>1</td> <td>RI -></td> <td>DO</td> </tr> <tr> <td>2</td> <td>DI -></td> <td>RO</td> </tr> <tr> <td>3</td> <td>DI -></td> <td>DO</td> </tr> </table>	1	RI ->	DO	2	DI ->	RO	3	DI ->	DO	DO [0]
1	RI ->	DO										
2	DI ->	RO										
3	DI ->	DO										
[TYPE]		[SELECT]	ENABLE DISABLE									

- 6** Select the kind of redirection you want:

- To redirect RI to DO, select 1, RI->DO.
- To redirect DI to RO, select 2, DI->RO.
- To redirect DI to DO, select 3, DI->DO.

NOTE RI and RO signal numbers cannot be changed.

- DI → RO** If you select DI → RO, you will see a screen similar to the following.

INTER CONNECT		JOINT 100%	
			1/24
No.	Enb/Disabl	INPUT	OUTPUT
1	ENABLE	DI [0] ->	RO [1]
2	DISABLE	DI [0] ->	RO [2]
3	DISABLE	DI [0] ->	RO [3]
4	DISABLE	DI [0] ->	RO [4]
5	DISABLE	DI [0] ->	RO [5]
6	DISABLE	DI [0] ->	RO [6]
24	DISABLE	DI [0] ->	RO [24]
[TYPE]		[SELECT]	ENABLE DISABLE

DI → DO If you select DI → DO, you will see a screen similar to the following.

INTER CONNECT		JOINT 100%	
		1/24	
No.	Enb/Disabl	INPUT	OUTPUT
1	ENABLE	DI [0] ->	DO [0]
2	DISABLE	DI [0] ->	DO [0]
3	DISABLE	DI [0] ->	DO [0]
4	DISABLE	DI [0] ->	DO [0]
5	DISABLE	DI [0] ->	DO [0]
6	DISABLE	DI [0] ->	DO [0]
24	DISABLE	DI [0] ->	DO [0]
[TYPE]		[SELECT]ENABLE DISABLE	

SI → DO If you select SI → DO, you will see a screen similar to the following.

INTER CONNECT		JOINT 100%	
		1/8	
No.	Enb/Disabl	INPUT	OUTPUT
1	ENABLE	SI [0] ->	DO [0]
2	DISABLE	SI [1] ->	DO [1] ← RESET
3	DISABLE	SI [2] ->	DO [0]
4	DISABLE	SI [3] ->	DO [0]
5	DISABLE	SI [4] ->	DO [3] ← CE-1
6	DISABLE	SI [5] ->	DO [4] ← CE-2
7	DISABLE	SI [6] ->	DO [2] ← START
8	DISABLE	SI [7] ->	DO [0]
[TYPE]		[SELECT]ENABLE DISABLE	

NOTE The default DO numbers for RESET, CE-1 (MODE SELECT switch 1), CE-2 (MODE SELECT switch 2), and START are specified automatically. You can change these numbers if desired.

The relationship between the MODE SELECT switch signals and the modes of operation is shown in Table 3-2. The MODE SELECT switch is used only for the Control Reliable (RS-1/RS-4) option.

Table 3-2. Relationship Between the MODE SELECT Switch Signals and Modes of Operation

Signal	Mode of Operation		
	T2	T1	AUTO
CE-1	0	1	1
CE-2	0	0	1

ES → DO If you select ES → DO, you will see a screen similar to the following.

INTER CONNECT		JOINT 100%	
No.	Enb/Disabl	INPUT	OUTPUT
1	ENABLE	[EMGOP]	-> DO [31]
2	DISABLE	[EMGTP]	-> DO [29]
3	DISABLE	[DEADMAN]	-> DO [30]
4	DISABLE	[FENCE]	-> DO [30]
5	DISABLE	[ROT]	-> DO [0]
6	DISABLE	[HBK]	-> DO [0]
7	DISABLE	[EMGEX]	-> DO [33]
8	DISABLE	[PPABN]	-> DO [0]
9	DISABLE	[BELTBREAK]	-> DO [0]
10	DISABLE	[FALM]	-> DO [0]
11	DISABLE	[SVON]	-> DO [0]
12	DISABLE	[IMSTP]	-> DO [32]
13	DISABLE	[BRKHL D]	-> DO [0]
14	DISABLE	[USRALM]	-> DO [0]

1/14

[TYPE] [SELECT] ENABLE DISABLE

← SOP E-STOP
← TP E-STOP
← DEADMAN
← FENCE OPEN
← EXTERNAL E-STOP
← UOP E-STOP

NOTE Refer to the *FANUC Robotics SYSTEM R-J2 Controller i-Size and B-size Controller Maintenance Manual* for more information on emergency stop signals.

NOTE The default DO numbers for STOP E-STOP, TP E-STOP, DEADMAN, FENCE OPEN, EXTERNAL E-STOP, and UOP E-STOP are specified automatically. You can change these numbers if desired.

7 For each signal you want to redirect, enter the signal number of the DI or DO.

NOTE If the signal number of the DO or the DI is 0, **the signal will not be redirected.**

8 For each signal you want to redirect, enable or disable the redirection of the signal:

- **To enable the redirection**, press F4, ENABLE.
- **To disable the redirection**, press F5, DISABLE.

NOTE I/O interconnection changes take effect immediately. It is **NOT** necessary to turn the controller off then on for these changes to take effect.

NOTE The response time to update a signal is from 20 ms to 100 ms.

3.3 USER OPERATOR PANEL (UOP) I/O SIGNALS

The User Operator Panel (UOP) provides 18 input signals and 20 or 24 output signals (four are optional outputs), that can be connected to a remote device or a remote operator panel for control of the robot.

Most UOP I/O signals are active when the robot is in a remote condition—the remote switch on the operator panel is set to REMOTE. Signals that affect safety are always active.

For systems with a process I/O board, the UOP signals are already configured and assigned to dedicated ports. If you want to use UOP I/O and you do not have a process I/O board, you must configure UOP I/O. You can also:

- Control UOP outputs
- Add comments about UOP I/O

NOTE If you configure UOP input and output signals, the UI and UO physical locations are actually DI/DO on the Process I/O board or Modular I/O. In effect, the physical DI/DO can be double configured as both UI/UO and DI/DO logicals. This allows you to control or monitor UI/UO signals within your program by using the DI/DO instructions.

Configuring UOP I/O

Each signal must be configured to a rack, a slot in the rack, and the starting point for numbering when HandlingTool is loaded. See Figure 3-9 and Figure 3-10. You can change the configuration of:

- **Rack** – the physical location on which the input or output process I/O board or modular I/O is mounted. Your system can contain multiple racks. Process I/O boards are always assigned Rack 0. For modular I/O, the rack mounted closest to the main CPU is considered Rack 1.

NOTE For GEFANUC and Allen Bradley boards, the rack assignment is 16. The I/O assigned to the DeviceNet Interface is configured as racks 81 – 84.

- **Slot** – the space on the rack where the process I/O board or modular I/O is connected. For Process I/O boards, number from right to left. For modular I/O, number from left to right.

NOTE Allen Bradley and GEFANUC I/O boards, the slot assignment is 1. The slot number for DeviceNet Interface I/O is the MAC Id for the device.

- **Starting Point** – the physical position on the process I/O board or modular I/O of the first port to be included as UOP.

Figure 3-9. Process I/O Board Hardware Layout

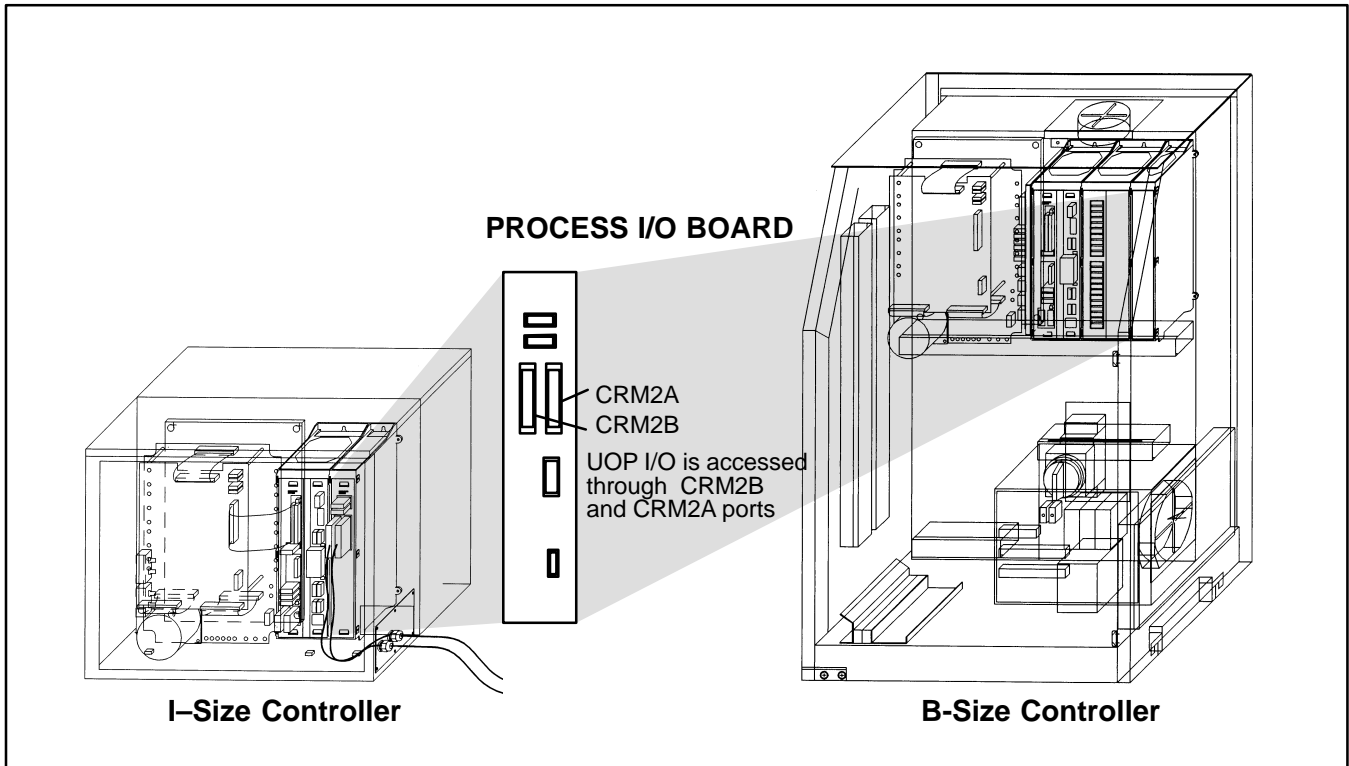
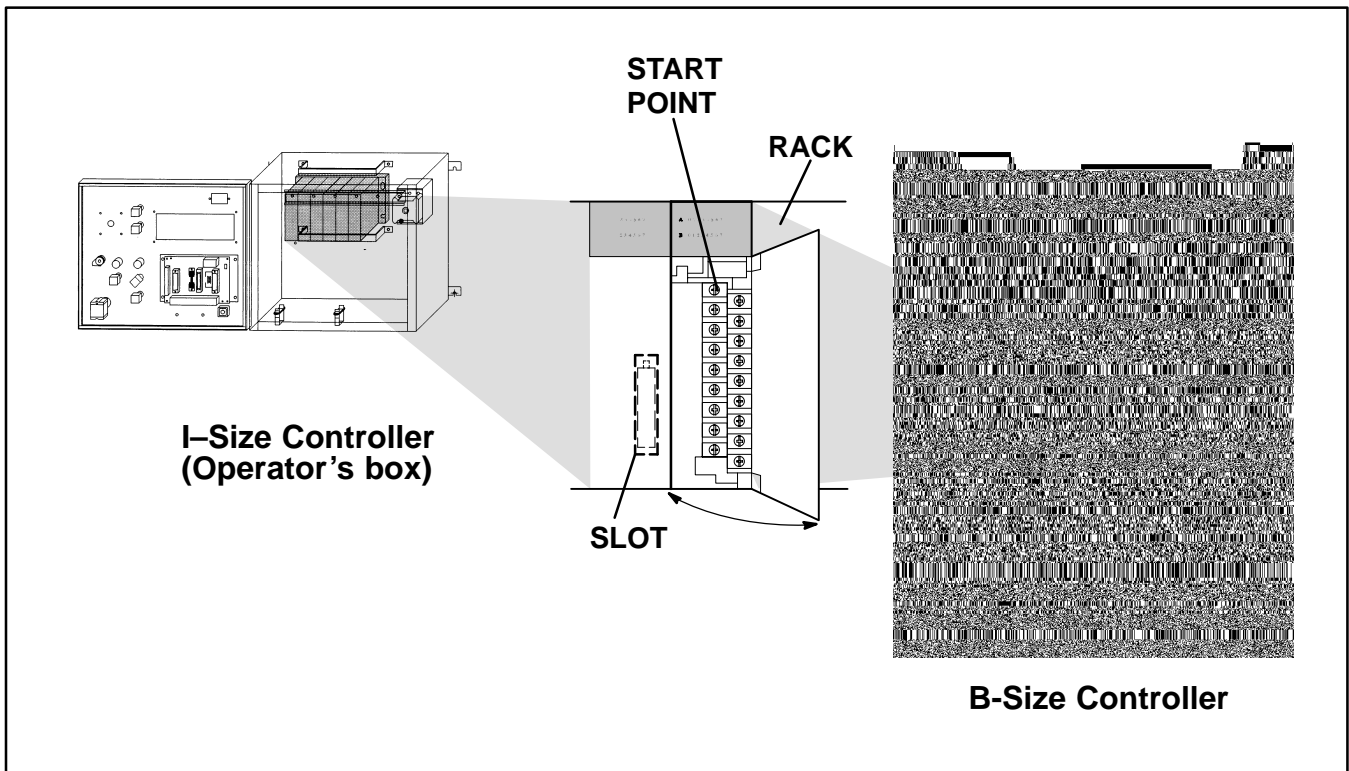


Figure 3-10. Modular I/O Hardware Layout



Controlling Outputs

Controlling outputs allows you to force a UOP signal manually.

Adding Comments About UOP I/O

Adding comments about UOP I/O allows you to include text that describes the signal. For example, you can add a comment to indicate the line that is physically connected to the port.

NOTE UOP I/O comments are installed by HandlingTool, but can be changed. Changing the comment does not change the function.

**3.3.1
UOP Input Signals**

For systems with a Process I/O board, the UOP signals are already configured and assigned to dedicated ports. The UOP input signals are listed and described in Table 3-3.

Table 3-3. UOP Input Signals

UOP INPUT SIGNAL	DESCRIPTION
<p>*IMSTP Always active UI[1]</p>	<p>*IMSTP is the immediate stop software signal. *IMSTP is a normally OFF signal held ON. When it is set to OFF, it</p> <ul style="list-style-type: none"> ● Pauses a program if one is running ● Immediately stops the robot and applies robot brakes ● Shuts off power to the servos <p>Error code SRVO-037 *IMSTP Input (Group:i) will be displayed when this signal is lost. This signal is always active.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>WARNING *IMSTP is a software controlled input and cannot be used for safety purposes. Use *IMSTP with EMGIN1 and EMGIN2 to use this signal with a hardware controlled emergency stop. Refer to the maintenance manual for connection information of EMGIN1 and EMG2.</p> </div>
<p>*HOLD Always active UI[2]</p>	<p>*HOLD is the external hold signal. *Hold is a normally OFF signal, held ON. When it is set to OFF, it will do the following:</p> <ul style="list-style-type: none"> ● Pause program execution ● Slow motion to a controlled stop and hold ● Optional Brake on Hold shuts off servo power after the robot stops
<p>*SFSPD Always active UI[3]</p>	<p>*SFSPD is the safety speed input signal. This signal is usually connected to the safety fence. *SFSPD is a normally OFF signal held ON. When it is set to OFF it will do the following:</p> <ul style="list-style-type: none"> ● Pause program execution ● Reduce the speed override value to that defined in a system variable. This value cannot be increased while *SFSPD is OFF. ● Display error code message SYST009. ● Not allow a REMOTE start condition. Start inputs from UOP or SOP are disabled when SFSPD is set to OFF and only the teach pendant has motion control with the speed clamped.

Table 3-3. (Cont'd) UOP Input Signals

UOP INPUT SIGNAL	DESCRIPTION
CSTOPI Always active UI[4]	<p>CSTOPI is the cycle stop input. The function of this signal depends on the system variable \$SHELL_CFG.\$USE_ABORT.</p> <p>If the system variable \$SHELL_CFG.\$USE_ABORT is set to FALSE, the CSTOPI input</p> <ul style="list-style-type: none"> • Clears the queue of programs to be executed that were sent by RSR signals <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>WARNING When \$SHELL_CFG.\$USE_ABORT is set to FALSE, CSTOPI does not immediately stop automatic program execution.</p> </div> <ul style="list-style-type: none"> • Automatic execution will be stopped after the current program has finished executing. <p>If the system variable \$SHELL_CFG.\$USE_ABORT is set to TRUE, the CSTOPI input</p> <ul style="list-style-type: none"> • Clears the queue of programs to be executed that were sent by RSR signals • Immediately aborts the currently executing program for programs that were sent to be executed by either RSR or PNS.
FAULT_RESET Always active UI[5]	<p>FAULT_RESET is the external fault reset signal. When this signal is received the following will happen:</p> <ul style="list-style-type: none"> • Error status is cleared • Servo power is turned on • The paused program will not be resumed
START Active when the robot is in a remote condition (CMDENBL = ON) UI[6]	<p>START is the remote start input. The function of this signal depends on the system variable \$SHELL_CFG.\$CONT_ONLY.</p> <p>If the system variable \$SHELL_CFG.\$CONT_ONLY is set to FALSE the START input signal</p> <ul style="list-style-type: none"> • Resumes a paused program • If a program is aborted, the currently selected program starts from the position of the cursor. <p>If the system variable \$SHELL_CFG.\$CONT_ONLY is set to TRUE the START input signal</p> <ul style="list-style-type: none"> • Resumes a paused program only. The PROD_START input must be used to start a program from the beginning.
HOME Active when the robot is in a remote condition UI[7]	<p>This signal can be defined to invoke a Macro program. The Macro program can move the robot to its HOME position.</p>
ENBL Always active UI[8]	<p>ENBL is the enable input. This signal must be ON to have motion control ability. When this signal is OFF, robot motion cannot be done. When ENBL is ON and the REMOTE switch on the operator panel is in the REMOTE position, the robot is in a remote operating condition.</p> <p>Display error code message SYST016.</p>
RSR 1-4 Active when the robot is in a remote condition (CMDENBL = ON) UI[9-12]	<p>RSR 1-4 are the robot service request input signals. When one of these signals is received, the corresponding RSR program is executed or, if a program is running currently, stored in a queue for later execution. RSR signals are used for production operation and can be received while an ACK output is being pulsed. See Figure 3-11.</p>

Table 3-3. (Cont'd) UOP Input Signals

UOP INPUT SIGNAL	DESCRIPTION
<p>PNS 1-8 Active when the robot is in a remote condition (CMDENBL = ON) UI[9-16]</p>	<p>PNS 1-8 inputs are program number select input signals. PNS selects programs for execution, but does not execute programs. Programs that are selected by PNS are executed using the START input or the PROD_START input depending on the value of the system variable \$SHELL_CFG.\$CONT_ONLY. Coordinate with CYCLE START.</p> <p>The PNS number is output by using the SNO signal (selected number output) and the SNACK signal (selected number acknowledge) will be pulsed. See Figure 3-12.</p>
<p>PNSTROBE Active when the robot is in a remote condition (CMDENBL = ON) UI[17]</p>	<p>PNSTROBE input is the program number select strobe input signal. See Figure 3-12.</p>
<p>PROD_START Active when the robot is in a remote condition (CMDENBL = ON) UI[18]</p>	<p>Production Start Input when used with PNS will initiate execution of the selected program from the PNS lines. When used without PNS, PROD_START executes the selected program from the current cursor position. Coordinate with CYCLE START. See Figure 3-12.</p>

*A normally OFF signal held ON. When it is set to OFF, certain conditions will result. Refer to the UOP signal definitions.

Figure 3-11 and Figure 3-12 provide information about the timing of the signals used with RSR and PNS.

Figure 3-11. RSR Timing Diagram

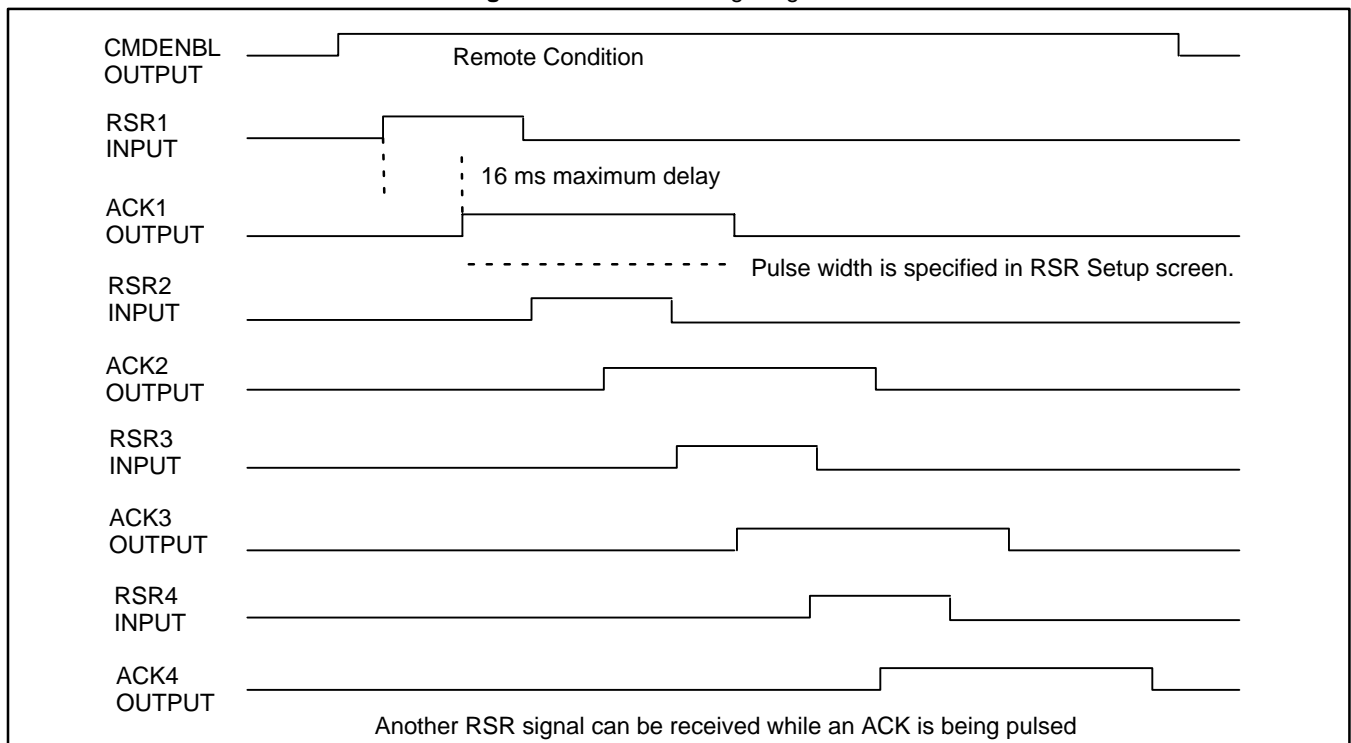
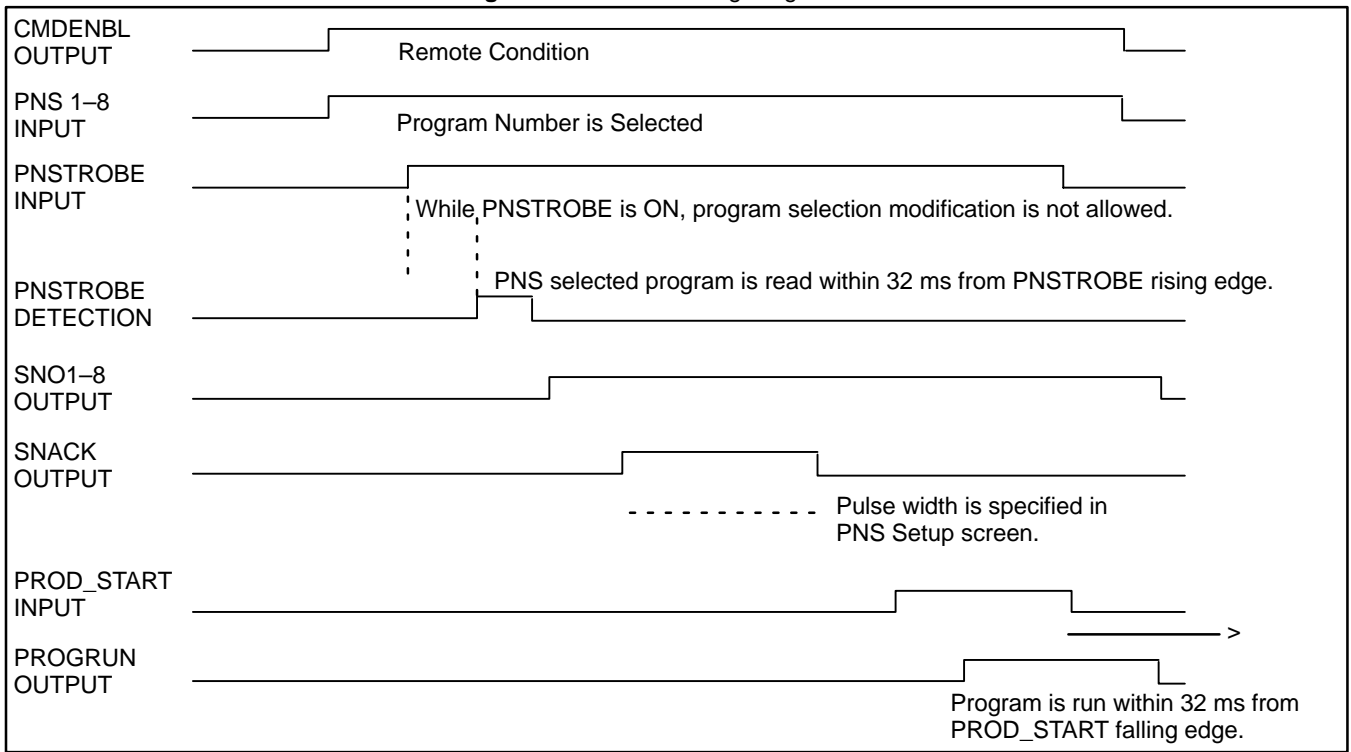


Figure 3-12. PNS Timing Diagram



3.3.2 UOP Output Signals

For systems with a Process I/O board, the UOP signals are already configured and assigned to dedicated ports. The UOP has the output signals that are listed and described in Table 3-4.

Table 3-4. UOP Output Signals

UOP OUTPUT SIGNAL	DESCRIPTION
CMDENBL UO[1]	CMDENBL is the command enable output. This output indicates that the robot is in a remote condition. This signal goes on when the remote switch is turned to ON. This output only stays on when the robot is not in a fault condition. When SYSRO is OFF, CMDENBL is OFF. See Figure 3-11 and Figure 3-12. This signal goes on when the following conditions are all satisfied . <ul style="list-style-type: none"> • Teach pendant disabled • Remote switch is ON • SFSPD input is ON • ENBL input is ON • \$RMT_MASTER system variable is 0 • Not in single step mode • Mode selection switch is set to AUTO (when mode select switch is installed)
SYSRDY UO[2]	SYSRDY is the system ready output. This output indicates that servos are turned on.
PROGRUN UO[3]	PROGRUN is the program run output. This output turns on when a program is running. See Figure 3-11.
PAUSED UO[4]	PAUSED is the paused program output. This output turns on when a program is paused.
HELD UO[5]	HELD is the hold output. This output turns on when the SOP HOLD button has been pressed, or the UOP *HOLD input is OFF.
FAULT UO[6]	FAULT is the error output. This output turns on when a program is in an error condition.
ATPERCH UO[7]	ATPERCH is the at perch output. This output turns on when the robot reaches the predefined perch position. When \$SHELL_WRK.\$KAREL_UOP=FALSE, then the system sets \$ATPERCH. The ATPERCH position = Reference position #1 .
TPENBL UO[8]	TPENBL is the teach pendant enable output. This output turns on when the teach pendant is on.
BATALM UO[9]	BATALM is the battery alarm output. This output turns on when the CMOS RAM battery voltage goes below 2.6 volts.
BUSY UO[10]	BUSY is the processor busy output. This signal turns on when the robot is executing a program or when the processor is busy.
ACK 1-4 UO[11-14]	ACK 1-4 are the acknowledge signals output 1 through 4. These signals turn on when the corresponding RSR signal is received. See Figure 3-11.
SNO 1-8 UO[11-18]	SNO 1-8 are the signal number outputs. These signals carry the 8 bit representation of the corresponding PNS selected program number. If the program cannot be represented by an 8 bit number, the signal is set to all zeroes or off. See Figure 3-12.
SNACK UO[19]	SNACK is the signal number acknowledge output. This output is pulsed if the program is selected by PNS input. See Figure 3-12.

*A normally OFF signal held ON. When it is set to OFF, certain conditions will result. Refer to the UOP signal definitions in Table 5-4 above.

Use Procedure 3-7 to configure UOP I/O – rack, slot and start point.

Procedure 3-7 Configuring UOP I/O – Rack, Slot, Start Point

- Step**
- 1 Press MENUS.
 - 2 Select I/O.
 - 3 Press F1, [TYPE].
 - 4 Select UOP. You will see either the UOP input or UOP output screens. See the following screen for an example.

I/O UOP Out		JOINT 50 %	
#	STATUS		
UO [1]	OFF	[Cmd enabled]
UO [2]	OFF	[System ready]
UO [3]	OFF	[Prg running]
UO [4]	OFF	[Prg paused]
UO [5]	OFF	[Motion held]
UO [6]	OFF	[Fault]
UO [7]	OFF	[At Perch]
UO [8]	OFF	[TP enabled]
UO [9]	OFF	[Batt alarm]
UO [10]	OFF	[Busy]
UO [11]	OFF	[ACK1/SN01]
UO [12]	OFF	[ACK2/SN02]
UO [13]	OFF	[ACK3/SN03]
UO [14]	OFF	[ACK4/SN04]
UO [15]	OFF	[SN05]
UO [16]	OFF	[SN06]
UO [17]	OFF	[SN07]
UO [18]	OFF	[SN08]
UO [19]	OFF	[SNACK]
UO [20]	OFF	[Reserved]
[TYPE]	CONFIG	IN/OUT	ON OFF

To change between the display of the input and output screens, press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down and up arrow keys.

- 5 Select the UOP signal you want to configure.

6 Press F2, CONFIG. You will see a screen similar to the following.

I/O UOP Out				JOINT	50 %
RANGE	RACK	SLOT	START PT	1/4	
UO [1- 8]	0	0	0		
UO [9- 16]	0	0	0		
UO [17- 24]	0	0	0		
UO [25- 32]	0	0	0		
[TYPE]	MONITOR	IN/OUT	DETAIL	HELP	

7 Configure the I/O:

- Move the cursor to RACK, type the new value, and press ENTER.
- Move the cursor to SLOT, type the new value, and press ENTER.
- Move the cursor to START PT, type the new value, and press ENTER.

8 To add a comment:

- Press F4, DETAIL. You will see a screen similar to the following.

I/O UOP Out				JOINT	50 %
UOP Output Detail				1/11	
UOP Output: UO [1]					
UOP Outputs: [1 - 8]					
1	Rack Number:	0			
2	Slot Number:	0			
3	Starting Point:	0			
4	Comment:	[1]	[
5	Comment:	[2]	[
6	Comment:	[3]	[
[TYPE]	NEXT	IN/OUT	>		

- Move the cursor to Comment, and press ENTER.
- Select a method of naming the comment.
- Press the appropriate function keys to add the comment.
- When you have finished, press ENTER.

⚠ WARNING

You must either verify the assignment or exit the detail menu by using the PREV key for the assignments to be recorded; otherwise, injury or damage to equipment could occur.

- 9 To determine if the assignment is valid**, press NEXT, >, and then press F2, VERIFY.
- If the assignment is valid, the message, “Port assignment is valid,” is displayed.
 - If the assignment is not valid, the message, “Port assignment is invalid,” is displayed.

 **CAUTION**

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

- Saving I/O Information** **10** To save the information (when all I/O is configured):

NOTE Make sure the UOP I/O menu is displayed.

- a Press MENUS.
- b Select FILE.
- c Press F1, [TYPE].
- d Select File.
- e Press F5, [UTIL].
- f Select Set Device.
- g Move the cursor to the device you want and press ENTER.
- h Press FCTN.
- i Select SAVE. The file will be saved to the DIOCFGSV.IO file on the default device.

 **WARNING**

You must turn off the controller and then turn it back on to use the new information; otherwise, you could injure personnel or damage equipment.

- 11** Turn off the controller. Turn on the controller so it can use the new information.

3.4 MODEL B I/O SETUP

Inputs and outputs (I/O) are electrical signals that enable the controller to communicate with the robot and external devices. This section contains information on how to set up I/O for Model B I/O modules. See Figure 3-13 and Figure 3-14.

NOTE Refer to Section 3.1 for information on how to set up process I/O boards and Model A modular I/O.

You must do the following to use distributed I/O:

1. Configure the distributed I/O DIP switches. Refer to Section 3.4.1.
2. Set up each basic digital I/O module. Refer to Section 3.4.2.
3. Set up user I/O signals. Refer to Sections 3.4.3 through 3.4.5.

Figure 3-13. Model B I/O – i-size Controller

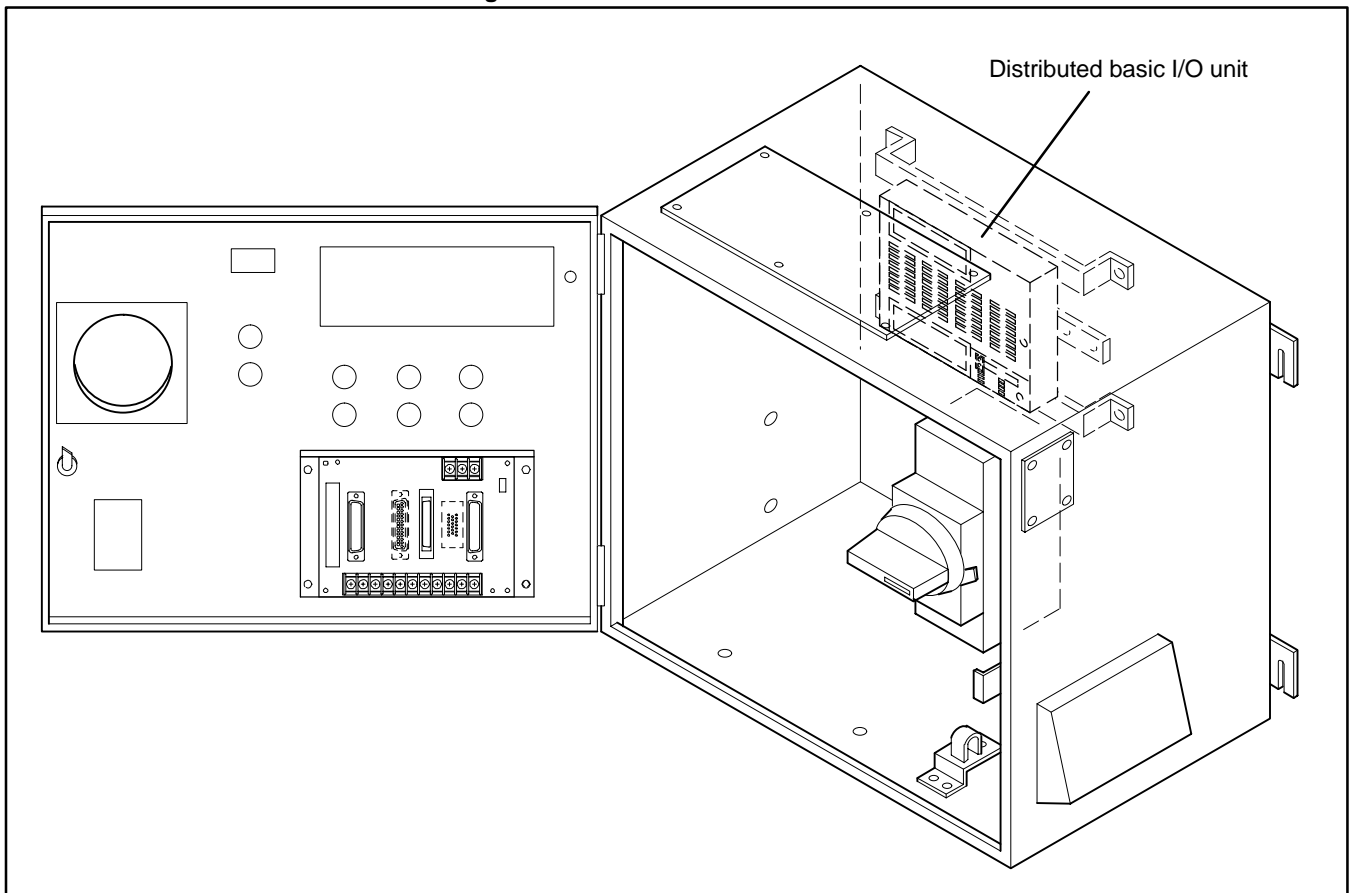
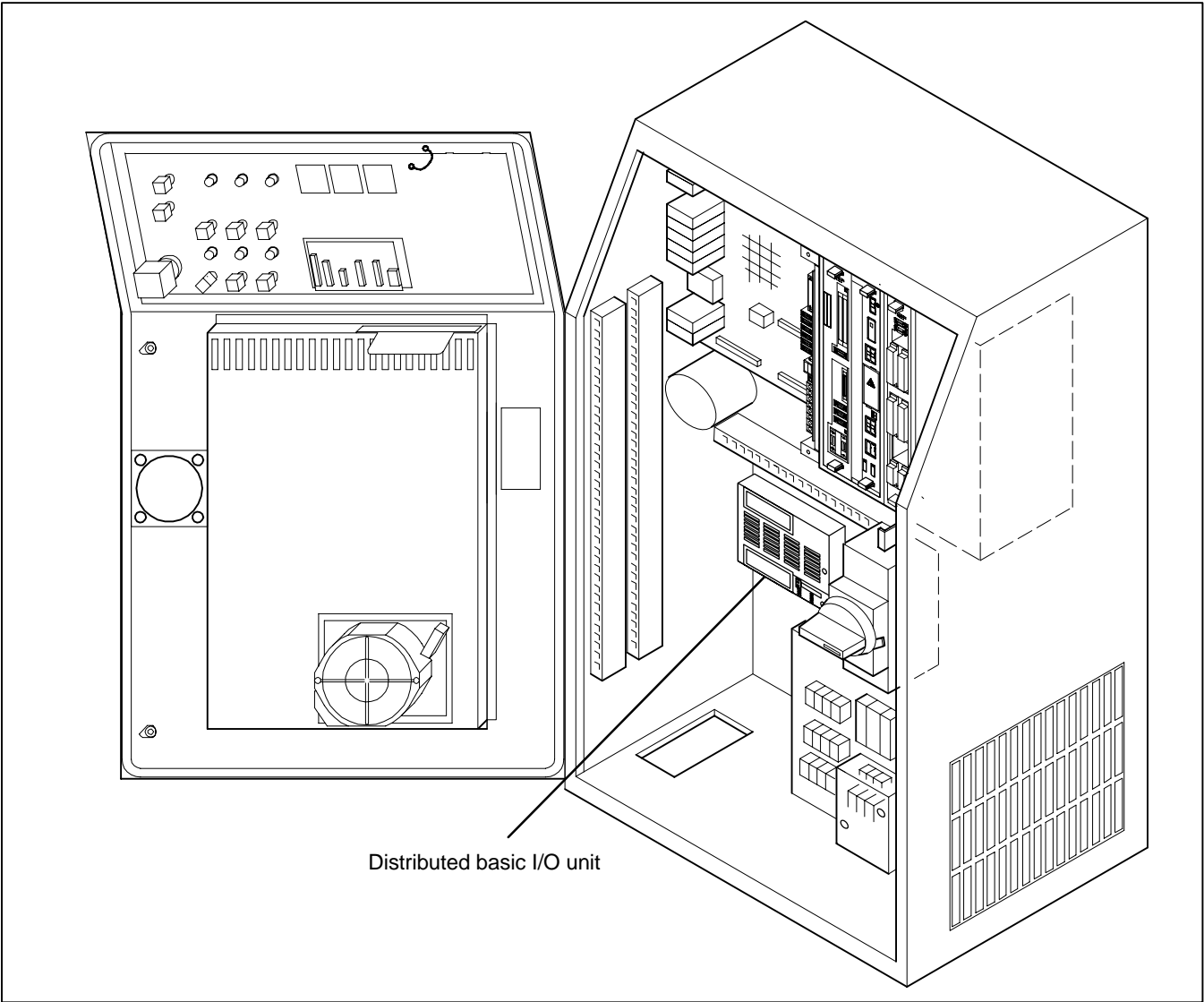


Figure 3-14. Model B I/O – B-size Controller



The following example describes each step of a typical distributed I/O setup.

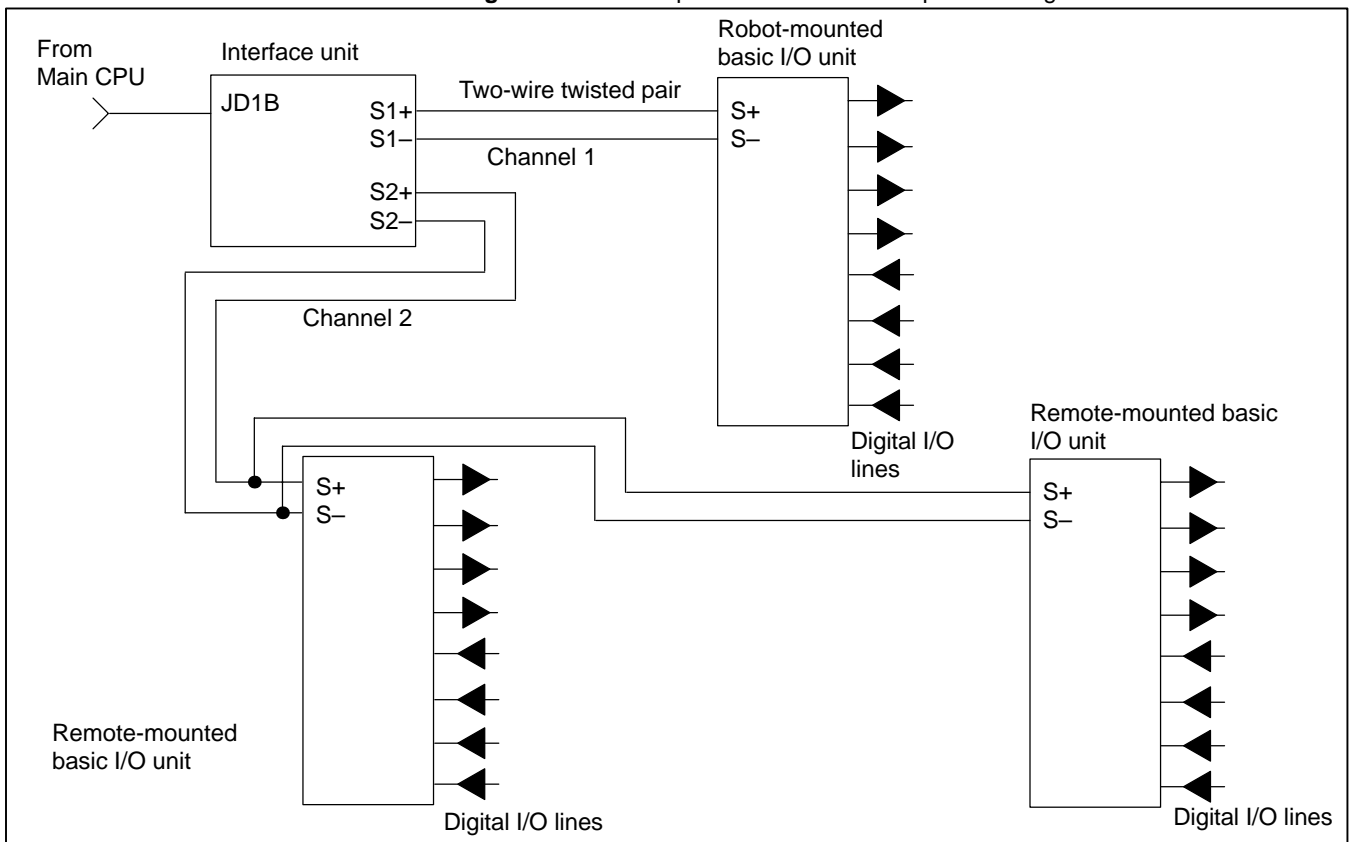
Model B I/O Example Setup

The examples in this section assume that you are setting up an installation with the distributed I/O interface unit mounted in the robot controller and three basic digital I/O units which can be mounted in various remote locations, such as:

- Robot arm (basic unit 1, connected to channel 1)
- Inside the operator box (basic unit 2, connected to channel 2)
- Inside a peripheral device (basic unit 3, connected to channel 2)

See Figure 3-15 for an illustration of this example setup.

Figure 3-15. Example Distributed I/O Setup Block Diagram



3.4.1 Setting the DIP Switches

You must set the following distributed I/O DIP switches:

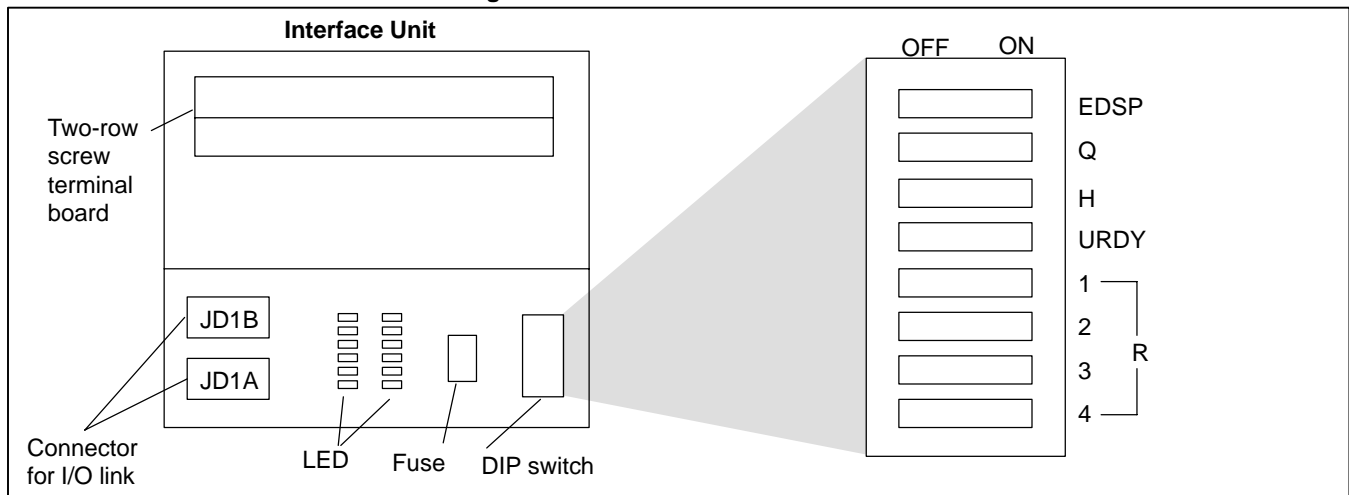
- On the interface unit, refer to Procedure 3-8
- On each basic digital I/O unit, refer to Procedure 3-9

Procedure 3-8 Setting the DIP Switches on the Interface Unit

Condition ■ The I/O modules are installed and wired properly.

Step 1 Locate the DIP switches on the interface unit. An eight-switch DIP switch package is mounted at the lower right corner of the interface module. See Figure 3-16.

Figure 3-16. Interface Unit DIP Switches



2 Set the EDSP switch to the ON position.

3 Set the communication speed using switches Q and H.

The I/O system can communicate at the following data rates:
1.2 Mbps, 600 Kbps, 300 Kbps.

Normally, you will use 1.2 Mbps (1.2 million bits per second).
However, when the total length of the communication lines exceeds
100 meters, a slower speed must be used.

Use the information in Table 3-5 to set switches Q and H.

Table 3-5. Communication Speed Settings for Switches Q and H

Q	H	Communication Speed
OFF	OFF	1.2 Mbps
OFF	ON	600 Kbps
ON	OFF	300 Kbps

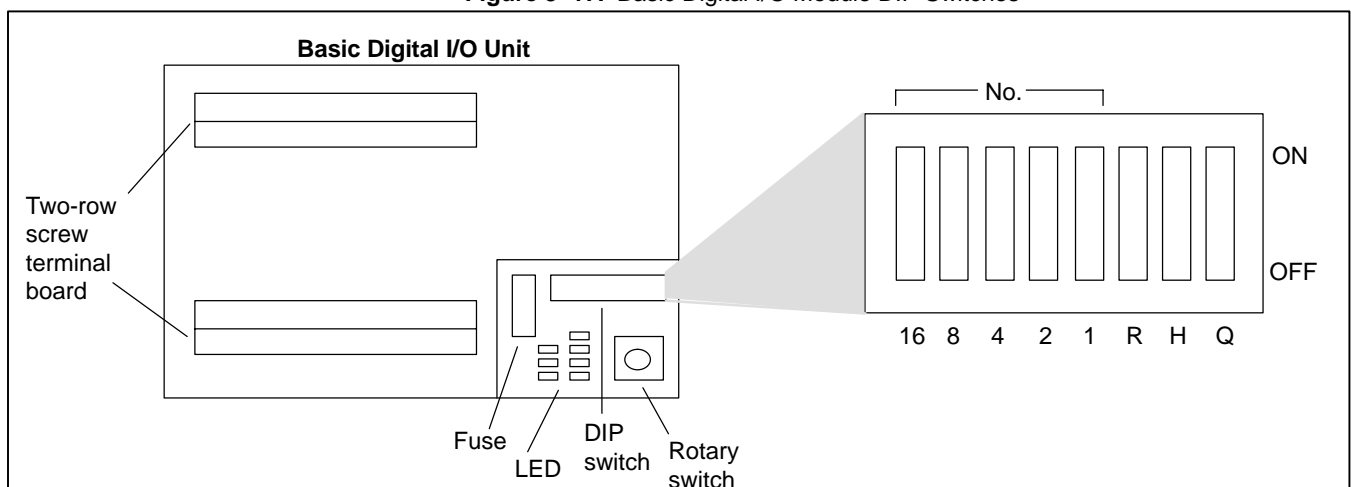
- 4 Write down the positions of switches Q and H. You will need this information when you configure the basic digital I/O units in Procedure 3-9 .
- 5 Set URDY to the OFF position.
- 6 Set the termination resistors, represented by switches R1 through R4.
 - a Examine the terminals for channel 1 (S1+ and S1-) and set switch R1 as follows:
 - **If one twisted-pair cable** is attached to these terminals, set the switch to ON.
 - **If more than one twisted-pair cable** is attached to these terminals, set the switch to OFF.
 - **If no wires** are attached to these terminals, the switch can be set to either ON or OFF.
 - b Examine the terminals for channel 2 (S2+ and S1-) and set switch R2 the same way you set switch R1 in Step 6a.
 - c Examine the terminals for channel 3 (S3+ and S3-) and set switch R3 the same way you set switch R1 in Step 6a.
 - d Examine the terminals for channel 4 (S4+ and S4-) and set switch R4 the same way you set switch R1 in Step 6a.

Procedure 3-9 Setting the DIP Switches on a Basic Digital I/O Unit

NOTE You must set the DIP switches for each basic digital I/O unit in your system.

- Condition** ■ The DIP switches on the interface unit have been set properly. (Procedure 3-8)
- Step** 1 Locate the DIP switches on the basic digital I/O unit. An eight-switch DIP switch package is mounted at the lower right corner of each basic digital I/O module. See Figure 3-17.

Figure 3-17. Basic Digital I/O Module DIP Switches



- 2** Set the unit number using switches 16, 8, 4, 2, and 1. These switches are set to show the number of the unit in binary notation. Refer to Table 3-6.

Table 3-6. Unit Number Settings of Switches 16, 8, 4, 2, and 1

Unit Number	Settings				
	16	8	4	2	1
1	OFF	OFF	OFF	OFF	ON
2	OFF	OFF	OFF	ON	OFF
3	OFF	OFF	OFF	ON	ON
4	OFF	OFF	ON	OFF	OFF
5	OFF	OFF	ON	OFF	ON
6	OFF	OFF	ON	ON	OFF
7	OFF	OFF	ON	ON	ON

- 3** Set the termination resistor, represented by switch R. Examine the terminals for S+ and S- and set switch R as follows:
- **If one twisted-pair cable** is attached to these terminals, set switch R to ON.
 - **If more than one twisted-pair cable** is attached to these terminals, set switch R to OFF.
 - **If no wires** are attached to these terminals, R can be set to either ON or OFF.

NOTE The positions of switches Q and H on the basic digital I/O module are reversed from the positions on the interface module. Be sure to set them properly.

- 4** Set the communication speed using switches Q and H. Use the same switch settings you used for the interface module in Procedure 3-8, Step 3.

3.4.2 Setting Up the Basic Digital I/O Units

You must set up each basic digital I/O unit you use. You do this from the I/O Link screen. Refer to Section 3.6 to set up Model B I/O basic digital I/O units.

3.4.3 Setting Up User I/O

After you have set up the DIP switches and have set up each basic digital I/O unit, you can set up user I/O. You can set up the following kinds of user I/O:

- **Digital** – DI[n] and DO[n]
- **Group** – GI[n] and GO[n]
- **Robot** - RI[n] and RO[n]
- **PLC** – PI[n] and PO[n]
- **SOP** – SI[n] and SO[n]
- **UOP** – UI[n] and UO[n]

These kinds of user I/O signals are attached to physical ports and are accessed from programs. The [n] corresponds to a signal number or group number. Setting up I/O establishes the correspondence between the signal number or group number and the physical port.

SOP inputs can be manually operated from the teach pendant and also monitored.

Robot inputs (RI) and outputs (RO) are preassigned. You cannot change the setup of RIs and ROs. Robot Input (RI) and Robot Output (RO) signals are on the Axis Control board located on the backplane of the controller. These signals interface to the end effector through a cable that is plugged into the base of the robot and are accessed through the EE (End Effector) plug on the robot arm.

The GEFanuc Genius I/O network and Allen Bradley Remote I/O are also available. For further information about these two kinds of I/O refer to *A User's Guide to the FANUC Robotics Remote I/O Interface for an Allen-Bradley PLC* or to *A User's Guide to the FANUC Robotics Genius Network Interface for GE Fanuc*.

3.4.4 Digital I/O

Digital I/O signals provide access to data on a single input or output signal line. Digital I/O signals can be ON or OFF.

NOTE If you configure UOP input and output signals, the UI and UO physical locations are actually **digital** I/O on the process I/O board or modular I/O. In effect, the physical digital I/O can be double configured as both user I/O and digital I/O logicals. This allows you to control or monitor user I/O signals within your program using the digital I/O instructions.

You can:

- Configure digital I/O
- Simulate digital I/O
- Control digital outputs
- Add comments about digital I/O

Complementary Output Signals

You can configure digital output signals to be controlled independently or in complementary pairs. If an output signal is controlled independently, a command to turn that output signal on or off controls only that output signal. If an output signal is controlled in a complementary pair, a command to turn that signal ON will also turn its pair OFF. A command to turn the signal OFF will also turn its pair ON.

Polarity

You can configure digital input/output signals with normal polarity (active ON) or inverse polarity (active OFF).

Configuring Digital I/O

Each signal is configured to a rack, a slot in the rack, and the starting point for numbering when HandlingTool is loaded. Digital I/O is configured in groups of eight. You can change the configuration of the

- **Rack** – varies depending on the kind of I/O you are using. Refer to Table 3-7. Your system can contain multiple racks.

Table 3-7. Rack Assignments for Different Kinds of I/O

Kind of I/O	Rack Assignment
Modular (Model A) I/O	Physical location on which the input or output modules are mounted. <ul style="list-style-type: none"> ● When used without distributed (Model B) I/O, begins at Rack 1. ● When used with distributed (Model B) I/O, the distributed system is Rack 1 and the modular rack is Rack 2.
Distributed (Model B) I/O	Rack 1
Process I/O	Rack 0
Allen-Bradley Remote I/O Interface	Rack 16
Genius Network Interface	Rack 16
DeviceNet Interface	Racks 81 – 84

- **Slot** – varies depending on the kind of I/O you are using. Refer to Table 3-8.

Table 3-8. Slot Assignments for Different Kinds of I/O

Kind of I/O	Slot Assignment
Modular (Model A) I/O	The space on the rack where the I/O module is connected.
Distributed (Model B) I/O	Begin at Slot 1 for the first unit
Process I/O	Begin at Slot 1 for the first unit
Allen-Bradley Remote I/O Interface	Slot 1
Genius Network Interface	Slot 1
DeviceNet Interface	The slot number is the MAC Id for the device.

- **Starting Point** – the physical position on the process I/O board or modular I/O board of the first port in a range of input or output signals. Valid starting points are 1, 9, 17, 25 and so forth.

On a distributed basic I/O unit that has both digital inputs and digital outputs, starting point 1 is used for both inputs and outputs.

Controlling Digital Outputs

Controlling outputs allows you to set the digital output value and turn it on in a program or to force it on manually. Digital I/O can be controlled individually.

Simulating Digital I/O

Simulating I/O allows you to test a program that uses I/O. Simulating I/O does not actually send output signals or receive input signals. Digital I/O can be simulated individually.

Adding Comments About Digital I/O

Adding comments about digital I/O allows you to include text that describes the signal. For example, you can add a comment to indicate the line that is physically connected to the port.

Use the following procedures to configure digital I/O:

- Use Procedure 3-10 to configure digital I/O – rack, slot and start point.
- Use Procedure 3-11 to configure digital I/O – polarity and complementary pairs.

Procedure 3-10 Configuring Digital I/O – Rack, Slot, Start Point

NOTE Digital I/O is configured by the system. Use this procedure if you want to change the configuration.

- Step**
- 1 Press I/O.
 - 2 Press F1, [TYPE].
 - 3 Select Digital. You will see either the digital input or digital output screens. See the following screen for an example.

I/O Digital In			JOINT	50 %
#	SIM	STATUS	1/256	
DI [1]	*	OFF	[]
DI [2]	*	OFF	[]
DI [3]	*	OFF	[]
DI [4]	*	OFF	[]
DI [5]	*	OFF	[]
DI [6]	*	OFF	[]
DI [7]	*	OFF	[]
DI [8]	*	OFF	[]
DI [9]	*	OFF	[]
DI [10]	*	OFF	[]

[TYPE] CONFIG IN/OUT SIMULATE UNSIM

To change between the display of the input and output screens, press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

- 4 Press F2, CONFIG. You will see a screen similar to the following.

I/O Digital In			JOINT	50 %
#	RANGE	RACK	SLOT	START PT
1	DI [1 - 8]	0	1	21
2	DI [9 - 16]	0	1	29
3	DI [17 - 24]	0	1	37
4	DI [25 - 32]	0	0	0
5	DI [33 - 40]	0	0	0
6	DI [41 - 48]	0	0	0
7	DI [49 - 56]	0	0	0
8	DI [57 - 64]	0	0	0
9	DI [65 - 72]	0	0	0

[TYPE] MONITOR IN/OUT DETAIL HELP >

[TYPE] VERIFY >

5 Configure the I/O:

- a** Move the cursor to RACK, type the value, and press ENTER.
- b** Move the cursor to SLOT, type the value, and press ENTER.
- c** Move the cursor to START PT, type the value, and press ENTER.

6 To add a comment:

- a** Press F4, DETAIL. You will see a screen similar to the following.

JOINT 50 %	
Digital Input Detail 1/19	
Digital Input: DI [1]	
Digital Inputs: [1 - 8]	
1	Rack Number: 0
2	Slot Number: 1
3	Starting Point: 21
4	Comment: [1] []
5	Comment: [2] []
6	Comment: [3] []
[TYPE]	NEXT IN/OUT >
[TYPE]	VERIFY >

- b** Move the cursor to the comment line and press ENTER.
 - c** Select a method of naming the comment.
 - d** Press the appropriate function keys to add the comment.
 - e** When you are finished, press ENTER.
- 7 To determine if the assignment is valid**, press NEXT, >, and then press F2, VERIFY.
- If the assignment is valid, the message, "Port assignment is valid," is displayed.
 - If the assignment is not valid, the message, "Port assignment is invalid," is displayed.

⚠ CAUTION

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving I/O Information

- 8** To save the information (when all I/O is configured):
 - a** Press **MENU**.
 - b** Select **FILE**.
 - c** Press **F1**, [**TYPE**].
 - d** Select **File**.
 - e** Press **F5**, [**UTIL**].
 - f** Select **Set Device**.
 - g** Move the cursor to the device you want and press **ENTER**.
 - h** Press **FCTN**.
 - i** Select **SAVE**. The file will be saved to the **DIOCFGSV.IO** file on the default device.

⚠ WARNING

You must turn off the controller and then turn it back on to use the new information; otherwise, you could injure personnel or damage equipment.

- 9** When you are finished configuring I/O, turn off the controller. Turn on the controller so it can use the new information.

Procedure 3-11 Configuring Digital I/O – Polarity and Complementary Pairs

- Step**
- 1 Press I/O.
 - 2 Press F1, [TYPE].
 - 3 Select Digital. You will see either the digital input or digital output screen. See the following screen for an example.

I/O Digital In				JOINT	50 %
	#	SIM	STATUS		1/256
DI [1]	*	OFF []
DI [2]	*	OFF []
DI [3]	*	OFF []
DI [4]	*	OFF []
DI [5]	*	OFF []
DI [6]	*	OFF []
DI [7]	*	OFF []
DI [8]	*	OFF []
DI [9]	*	OFF []
DI [10]	*	OFF []
[TYPE]	CONFIG	IN/OUT	SIMULATE	UNSIM	

To change the display between the Digital Input and Digital Output screen press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

- 4 Press F2, CONFIG. You will see a screen similar to the following.

I/O Digital In				JOINT	50 %
#	RANGE	RACK	SLOT	START PT	1/32
1 DI [1 - 8]	1	1	1	
2 DI [9 - 16]	1	1	9	
3 DI [17 - 24]	1	1	17	
4 DI [25 - 32]	1	1	25	
5 DI [33 - 40]	1	1	33	
6 DI [41 - 48]	*	*	*	
7 DI [49 - 56]	*	*	*	
8 DI [57 - 64]	*	*	*	
9 DI [65 - 72]	*	*	*	
[TYPE]	MONITOR	IN/OUT	DETAIL	HELP	>
[TYPE]	VERIFY				>

- 5 Move the cursor to the input or output group you want to configure.
- 6 Press F4, DETAIL.

7 To set polarity,12 Polarity:[1] [**INVERSE**]

- a** Move the cursor to the polarity of the signal you want to set. You will see a screen similar to the following.

JOINT 50 %			
Digital	Output	Detail	12/23
12 Polarity:	[1]	INVERSE	
13 Polarity:	[2]	NORMAL	
14 Polarity:	[3]	NORMAL	
15 Polarity:	[4]	NORMAL	
16 Polarity:	[5]	NORMAL	
17 Polarity:	[6]	NORMAL	
18 Polarity:	[7]	NORMAL	
19 Polarity:	[8]	NORMAL	
20 Complementary:	[1 - 2]	TRUE	
21 Complementary:	[3 - 4]	TRUE	
[TYPE]	NEXT	IN/OUT	INVERSE NORMAL >
[TYPE]	VERIFY		INVERSE NORMAL >

- b** Select the polarity you want:
- For inverse polarity, press F4, INVERSE.
 - For normal polarity, press F5, NORMAL.

8 To set complementary pairs (digital output signals only),20 Complementary:[1-2] **FALSE**]

- a** Move the cursor to the pair you want to set. You will see a screen similar to the following.

JOINT 50 %			
Digital	Output	Detail	20/23
14 Polarity:	[3]	NORMAL	
15 Polarity:	[4]	NORMAL	
16 Polarity:	[5]	NORMAL	
17 Polarity:	[6]	NORMAL	
18 Polarity:	[7]	NORMAL	
19 Polarity:	[8]	NORMAL	
20 Complementary	[1 - 2]	FALSE	
21 Complementary	[3 - 4]	TRUE	
22 Complementary	[5 - 6]	TRUE	
23 Complementary	[7 - 8]	TRUE	
[TYPE]	NEXT	IN/OUT	TRUE FALSE >
[TYPE]	VERIFY		TRUE FALSE >

- b** Select the complementary value:
- For no complementary pair, press F5, FALSE.
 - For a complementary pair, press F4, TRUE.

 **CAUTION**

When all I/O is configured, save the information to a storage device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving I/O Information

- 9** To save the information (when all I/O is configured):
 - a** Press **MENU**.
 - b** Select **FILE**.
 - c** Press **F1**, [**TYPE**].
 - d** Select **File**.
 - e** Press **F5**, [**UTIL**].
 - f** Select **Set Device**.
 - g** Move the cursor to the device you want and press **ENTER**.
 - h** Press **FCTN**.
 - i** Select **SAVE**. The file will be saved to the **DIOCFGSV.IO** file on the default device.

 **WARNING**

You must turn off the controller and then turn it back on to use the new information; otherwise, you could injure personnel or damage equipment.

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

3.4.5 Group I/O

Group I/O signals provide access to data on more than one input or output signal line at one time. Group I/O instructions allow a program to monitor or set a group of input or output signals as a binary number.

NOTE If you configure UOP input and output signals, the user I/O physical locations are actually digital I/O on the process I/O board, modular, or distributed I/O. In effect, the physical digital I/O can be double configured as both user I/O and digital I/O logicals.

If you then group your digital I/O signals, you can control or monitor user I/O signals within your program using the group I/O instructions. For example, you can configure the UOP signals into groups and issue a single command to control the entire group.

If you want to use group I/O, you must configure group I/O. You can also:

- Control group outputs
- Simulate group I/O
- Add comments about group I/O

Configuring Group I/O

Each group must be configured to a rack, a slot in the rack, the starting point for numbering, and the number of points when HandlingTool is loaded. You can change the configuration of:

- **Rack** – varies depending on the kind of I/O you are using. Refer to Table 3-9. Your system can contain multiple racks.

Table 3-9. Rack Assignments for Different Kinds of I/O

Kind of I/O	Rack Assignment
Modular (Model A) I/O	Physical location on which the input or output modules are mounted. <ul style="list-style-type: none"> ● When used without distributed (Model B) I/O, begins at Rack 1. ● When used with distributed (Model B) I/O, the distributed system is Rack 1 and the modular rack is Rack 2.
Distributed (Model B) I/O	Rack 1
Process I/O	Rack 0
Allen-Bradley Remote I/O Interface	Rack 16
Genius Network Interface	Rack 16
DeviceNet Interface	Racks 81 – 84

- **Slot** – varies depending on the kind of I/O you are using. Refer to Table 3-10.

Table 3-10. Slot Assignments for Different Kinds of I/O

Kind of I/O	Slot Assignment
Modular (Model A) I/O	The space on the rack where the I/O module is connected.
Distributed (Model B) I/O	Begin at Slot 1 for the first unit
Process I/O	Begin at Slot 1 for the first unit
Allen-Bradley Remote I/O Interface	Slot 1
Genius Network Interface	Slot 1
DeviceNet Interface	The slot number is the MAC Id for the device.

- **Starting Point** – the physical position on the process I/O board or modular I/O board of the first port in a range of input or output signals. Valid starting points are 1, 9, 17, 25 and so forth.
On a distributed basic I/O unit that has both digital inputs and digital outputs, starting point 1 is used for both inputs and outputs.
- **Number of Points** – indicates how many inputs or outputs will be in a group. The lowest number, or starting point, of the input or output is the least significant bit. The number of points can be from 1 up to and including 16.

Controlling Group Outputs

Controlling outputs allows you to set the group output value and turn it on in a program or to force it on manually.

Simulating Group I/O

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

Adding Comments About Group I/O

Adding comments about group I/O allows you to include text that describes the signal. For example, you can add a comment to indicate why you are grouping the signals.

Use Procedure 3-12 to configure group I/O – rack, slot, start point, and number of points.

Procedure 3-12 Configuring Group I/O – Rack, Slot, Start Point, Num Pts

- Step**
- 1 Press I/O.
 - 2 Press F1, [TYPE].
 - 3 Select Group. You will see either the group input or group output screens. See the following screen for an example.

I/O Group Out				JOINT	50 %
#	SIM	VALUE			1/25
GO [1]	*	0	[]
GO [2]	*	0	[]
GO [3]	*	0	[]
GO [4]	*	0	[]
GO [5]	*	0	[]
GO [6]	*	0	[]
GO [7]	*	0	[]
GO [8]	*	0	[]
GO [9]	*	0	[]
GO [10]	*	0	[]

[TYPE] CONFIG IN/OUT SIMULATE UNSIM

To change between the display of the input and output screens, press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

- 4 Press F2, CONFIG. You will see a screen similar to the following.

I/O Group Out					JOINT	50 %
GO #	RACK	SLOT	START PT	NUM	PTS	1/25
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	
5	0	0	0	0	0	
6	0	0	0	0	0	
7	0	0	0	0	0	
8	0	0	0	0	0	
9	0	0	0	0	0	

[TYPE] MONITOR IN/OUT DETAIL HELP >

[TYPE] VERIFY >

5 Configure the I/O:

- a** Move the cursor to RACK, type the value, and press ENTER.
- b** Move the cursor to SLOT, type the value, and press ENTER.
- c** Move the cursor to START PT, type the value, and press ENTER.
The starting point can be any number up to and including 999.
- d** Move the cursor to NUM PTS, type the value, and press ENTER.
The number of points can be from 1 up to and including 16.

6 To add a comment:

- a** Press F4, DETAIL. You will see a screen similar to the following.

I/O	Group	Out	JOINT	50	%
Group Output Detail					1/5
Group Output: GO [1]					
1	Rack Number:		0		
2	Slot Number:		0		
3	Starting Point:		0		
4	Number of Points:		0		
5	Comment:		█]		
[TYPE]	NEXT	IN/OUT	>		
[TYPE]	VERIFY	>			

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

7 To determine if the assignment is valid, press NEXT, >, and then press F2, VERIFY.

- If the assignment is valid, the message, "Port assignment is valid," is displayed.
- If the assignment is not valid, the message, "Port assignment is invalid," is displayed.

 CAUTION

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving I/O Information

- 8** To save the information (when all I/O is configured):
 - a** Press **MENU**.
 - b** Select **FILE**.
 - c** Press **F1**, [**TYPE**].
 - d** Select **File**.
 - e** Press **F5**, [**UTIL**].
 - f** Select **Set Device**.
 - g** Move the cursor to the device you want and press **ENTER**.
 - h** Press **FCTN**.
 - i** Select **SAVE**. The file will be saved to the **DIOCFGSV.IO** file on the default device.

 WARNING

You must turn off the controller and then turn it back on to use the new information; otherwise, you could injure personnel or damage equipment.

- 9** Turn off the controller. Turn on the controller so it can use the new information.

3.5 PLC I/O SETUP

PLC I/O is an option package. In addition to the feature of transferring I/O signal status information, you can configure your system to allow the cell controller (PLC) to control the modular and fixed discrete I/O within a controller directly. You do this by assigning two dedicated signal types, PI (PLC inputs) and PO (PLC outputs).

The cell controller views the robot I/O interface as a remote I/O rack (RIO). The use of the RIO interface along with the PLC I/O, outputs from the cell controller system become output from the R-J2 controller output modules, and inputs into the R-J2 controller input modules become inputs into the cell controller I/O system.

The dedicated signal types, PI and PO, can have index numbers from 1 to 128. These index numbers correspond directly to the 128 input and 128 output points on the RIO interface.

You can:

- Configure PLC I/O
- Add comments about PLC I/O

Complementary Output Signals

You can configure PLC output signals to be controlled independently or in complementary pairs. If an output signal is controlled independently, a command to turn that output signal on or off controls only that output signal. If an output signal is controlled in a complementary pair, a command to turn that signal on will also turn its pair off. A command to turn the signal off will also turn its pair on.

Polarity

You can configure PLC input/output signals with normal polarity (active ON) or inverse polarity (active OFF).

Configuring PLC I/O

PLC I/O is configured in groups of eight. You must assign the PLC I/O to a rack, a slot in the rack, and the starting point for numbering when HandlingTool is loaded. See Figure 3-18 and Figure 3-19. You can change the configuration of the

- **Rack** – the physical location on which the input or output process I/O board or modular I/O is mounted. Your system can contain multiple racks. Process I/O boards are always assigned Rack 0. Modular I/O begins at Rack 1. Valid numbers are (1-5).
- **Slot** – the space on the rack where the modular I/O module is connected. The slot number is also used to distinguish one process I/O board from another when more than one is used.
- **Starting Point** – the physical position on the process I/O board or modular I/O board of the first port in a range of input or output signals. Valid starting points are 1, 9, 17, 25 and so forth.

Figure 3-18. PLC I/O Process I/O Board Hardware Layout

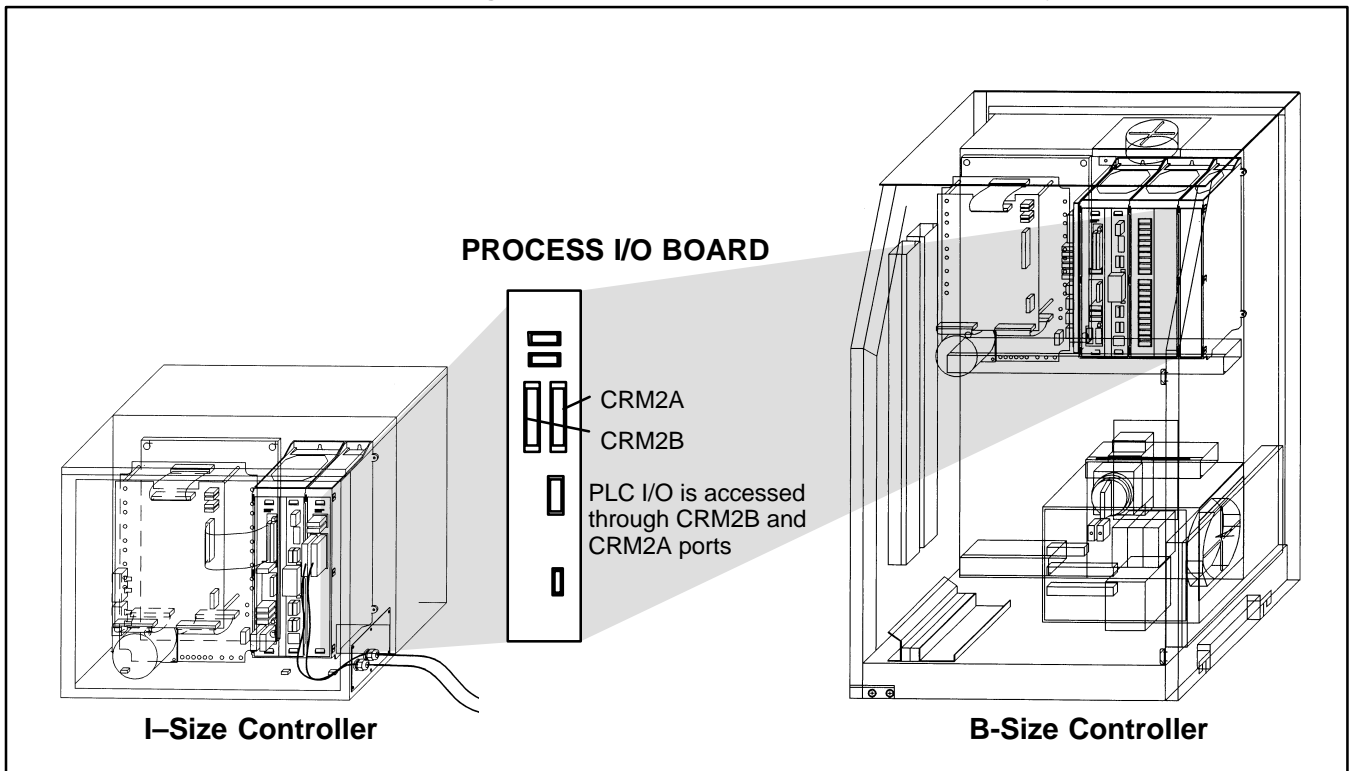
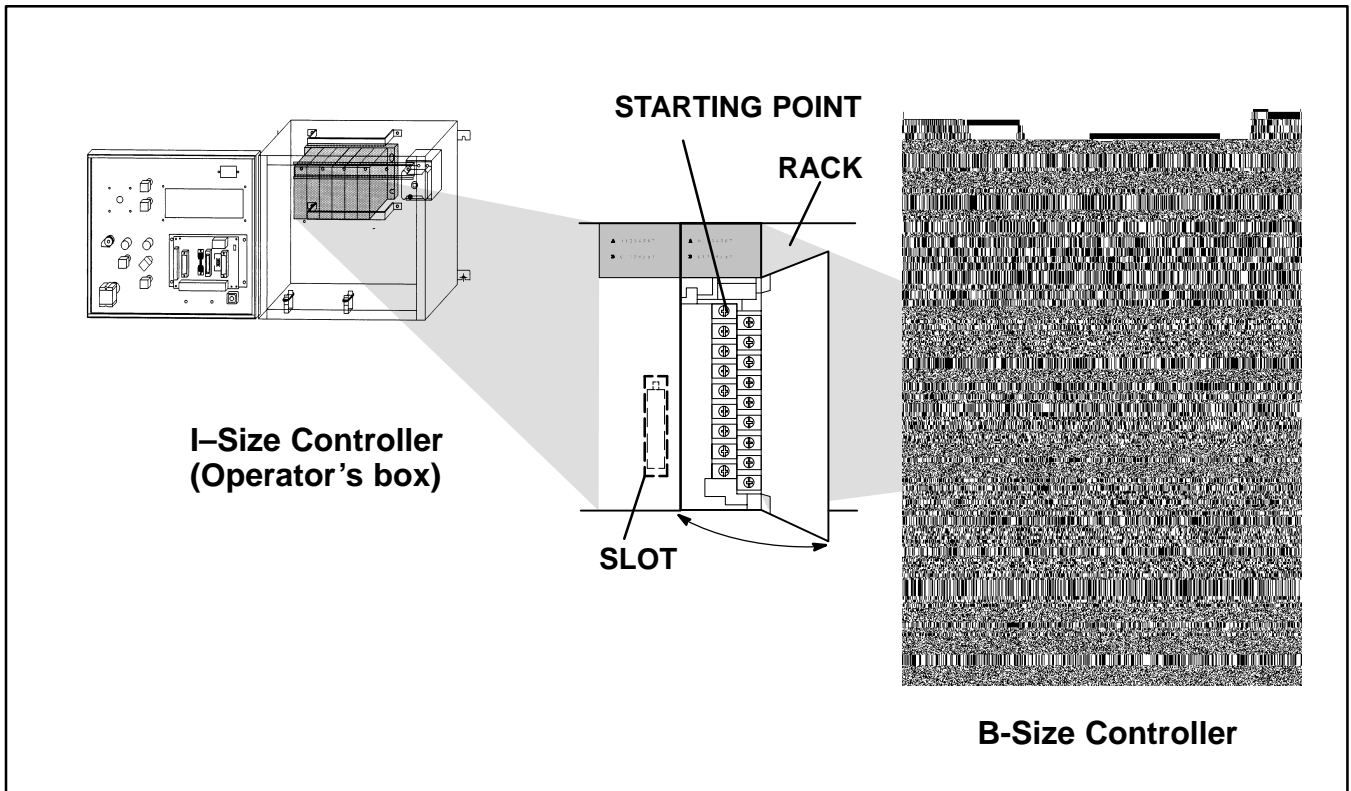


Figure 3-19. PLC I/O Modular I/O Hardware Layout



Adding Comments About PLC I/O

Adding comments about PLC I/O allows you to include text that describes the signal. For example, you can add a comment to indicate the line that is physically connected to the port.

Use Procedure 3-13 to configure PLC I/O – rack slot and start point.

Use Procedure 3-14 to configure PLC I/O – polarity and complementary pairs.

Procedure 3-13 Configuring PLC I/O – Rack, Slot, Start Point

- Step**
- 1 Press MENUS.
 - 2 Select I/O.
 - 3 Press F1, [TYPE]
 - 4 Select PLC. You will see either the PLC input or PLC output screens. See the following screen for an example.

I/O	PLC Out	#	SIM	STATUS	E1	JOINT	50 %
PO	[1]	*	OFF	[]
PO	[2]	*	OFF	[]
PO	[3]	*	OFF	[]
PO	[4]	*	OFF	[]
PO	[5]	*	OFF	[]
PO	[6]	*	OFF	[]
PO	[7]	*	OFF	[]
PO	[8]	*	OFF	[]
PO	[9]	*	OFF	[]
PO	[10]	*	OFF	[]
[TYPE]			CONFIG				IN/OUT

To change between the display of the input and output screens, press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

5 Press F2, CONFIG. You will see a screen similar to the following.

I/O PLC Out		E1	JOINT	50 %
#	RANGE	RACK	SLOT	1/21 START PT
1	PO [1 - 8]	0	0	0
2	PO [9 - 16]	0	0	0
3	PO [17 - 24]	0	0	0
4	PO [25 - 32]	0	0	0
5	PO [33 - 40]	0	0	0
6	PO [41 - 48]	0	0	0
7	PO [49 - 56]	0	0	0
8	PO [57 - 64]	0	0	0
9	PO [65 - 72]	0	0	0

[TYPE] MONITOR IN/OUT DETAIL HELP >

6 Configure the I/O:

- a Move the cursor to RACK, type the value, and press ENTER.
- b Move the cursor to SLOT, type the value, and press ENTER.
- c Move the cursor to START PT, type the value, and press ENTER.

7 To add a comment:

- a Press F4, DETAIL. You will see a screen similar to the following.

E1		JOINT	50 %
PLC Input Detail		1/23	
PLC Input: PI [1]			
PLC Inputs: [1 - 8]			
1	Rack Number:	0	
2	Slot Number:	0	
3	Starting Point:	0	
4	Comment:[1]	[]
5	Comment:[2]	[]
6	Comment:[3]	[]

[TYPE] NEXT IN/OUT >

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

- 8 To determine if the assignment is valid**, press NEXT, >, and then press F2, VERIFY.
- If the assignment is valid, the message, “Port assignment is valid,” is displayed.
 - If the assignment is not valid, the message, “Port assignment is invalid,” is displayed.

**WARNING**

You must turn off the controller and turn on the controller to use the new information; otherwise, you could injure personnel or damage equipment.

- 9** When you are finished configuring I/O, turn off the controller. Turn on the controller so it can use the new information.

Procedure 3-14 Configuring PLC I/O – Polarity and Complementary Pairs

- Step**
- 1 Press MENUS.
 - 2 Select I/O.
 - 3 Press F1, [TYPE].
 - 4 Select PLC. You will see either the PLC input or PLC output screen. See the following screen for an example.

I/O PLC Out		E1	JOINT	50 %
#	SIM	STATUS		
PO [1]	*	OFF []
PO [2]	*	OFF []
PO [3]	*	OFF []
PO [4]	*	OFF []
PO [5]	*	OFF []
PO [6]	*	OFF []
PO [7]	*	OFF []
PO [8]	*	OFF []
PO [9]	*	OFF []
PO [10]	*	OFF []
[TYPE]	CONFIG	IN/OUT	SIMULATE	UNSIM

To change the display between the input and output screen press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

5 Press F2, CONFIG. You will see a screen similar to the following.

I/O PLC Out		E1	JOINT	50 %
#	RANGE	RACK	SLOT	START PT
1	PO [1 - 8]	1	1	1
2	PO [9 - 16]	1	1	9
3	PO [17 - 24]	1	1	1
4	PO [25 - 32]	1	1	9
5	PO [33 - 40]	1	1	1
6	PO [41 - 48]	*	*	*
7	PO [49 - 56]	*	*	*
8	PO [57 - 64]	*	*	*
9	PO [65 - 72]	*	*	*

1/21

[TYPE] MONITOR IN/OUT DETAIL HELP >

The PI and PO signals under the range column refers to the I/O group of 8 in the overall group of PLC I/O from 1 to 128.

6 Move the cursor to the input or output group you want to configure.

7 Press F4, DETAIL.

8 To set polarity,

12 Polarity:[1] [INVERSE]

a Move the cursor to the polarity of the signal you want to set. You will see a screen similar to the following.

I/O PLC Out		E1	JOINT	50 %
PLC Output Detail			14/23	
14	Polarity:[2]	INVERSE		
15	Polarity:[2]	NORMAL		
16	Polarity:[3]	NORMAL		
17	Polarity:[4]	NORMAL		
18	Polarity:[5]	NORMAL		
19	Polarity:[6]	NORMAL		
21	Polarity:[7]	NORMAL		
22	Polarity:[8]	NORMAL		
23	Polarity:[9]	NORMAL		

[TYPE] NEXT IN/OUT INVERSE NORMAL >

b Select the polarity you want:

- For inverse polarity, press F4, INVERSE.
- For normal polarity, press F5, NORMAL.

9 To set complementary pairs (output signals only),20 Complementary:[1-2] **FALSE**]

- a** Move the cursor to the pair you want to set. You will see a screen similar to the following.

I/O	PLC Out	E1	JOINT	50 %
PLC Output Detail				20/23
14	Polarity:[4]	NORMAL		
15	Polarity:[5]	NORMAL		
16	Polarity:[6]	NORMAL		
17	Polarity:[7]	NORMAL		
18	Polarity:[8]	NORMAL		
19	Polarity:[9]	NORMAL		
20	Complementary[1-2]	FALSE		
21	Complementary[3-4]	TRUE		
22	Complementary[5-6]	TRUE		
23	Complementary[7-8]	TRUE		
[TYPE]	NEXT	IN/OUT	TRUE	FALSE >

- b** Select the complementary value:

- For no complementary pair, press F5, FALSE.
- For a complementary pair, press F4, TRUE.

⚠ WARNING

You must turn off the controller and turn the controller on to use the new information; otherwise, you could injure personnel or damage equipment.

⚠ CAUTION

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving I/O Information

- 10** To save the information (when all I/O is configured):

- Press **MENUS**.
- Select **FILE**.
- Press **F1**, **[TYPE]**.
- Select **File**.
- Press **F5**, **[UTIL]**.
- Select **Set Device**.
- Move the cursor to the device you want and press **ENTER**.
- Press **FCTN**.
- Select **SAVE**. The file will be saved to the **DIOCFGSV.IO** file on the default device.

- 11** Turn off the controller. Turn on the controller so it can use the new information.

3.6 I/O LINK SCREEN

The function of this screen is to set up Model B I/O unit and display the configuration of I/O link devices.

I/O link screen consists of the following screens:

- I/O link device screen, Section 3.6.1
- Model B I/O detail screen, Section 3.6.2
- Number of ports setting screen, Section 3.6.3

3.6.1 I/O Link Device Screen

This screen lists all process I/O boards, model A I/O racks, model B interface units, and PLC like devices connected to the controller through the I/O-LK connector on the MAIN CPU printed circuit board.

The following is the example of I/O link device screen when Process I/O board CB is connected to JD1A of R-J2 controller, one I/O unit Model B interface, and two I/O unit Model A racks are connected.

I/O Link Device		JOINT	100%
			1 / 4
Device Name	Comment		RackSlot
1 PrcI/O AA [] 0	1
2 Model B [] 1	0
3 Model A [] 2	0
4 Model A [] 3	0
[TYPE]	DETAIL		ASG_CLR

To display this screen, press I/O, F1, [TYPE], and select Link Device.

Table 3-11 contains descriptions of the device names displayed on the I/O Link Device screen.

Table 3-11. Device Names

Device name displayed	Device Description
Prcl/O AA	Process I/O board AA
Prcl/O AB	Process I/O board AB
Prcl/O BA	Process I/O board BA
Prcl/O BB	Process I/O board BB
Prcl/O CA	Process I/O board CA
Prcl/O CB	Process I/O board CB
Prcl/O DA	Process I/O board DA
Laser	Laser I/O
MODEL A	FANUC I/O UNIT MODEL A
MODEL B	FANUC I/O UNIT MODEL B
90-30 PLC	GEFanuc 90-30 PLC slave mode interface unit
I/O adptr	I/O Link adapter
JEMA PC	JEMA PC
R-J2 Mate	R-J2 Mate slave mode
Weld I/F	Weld I/F board
Unknown	Controller does not know the ID of this device

The slot value of Model A and Model B on this screen is 0.

For devices whose number of ports cannot be decided automatically, you can use the detail screen to set the number of ports manually. See Procedure 3-15 .

The devices that have access to the detail screen are listed in Table 3-12.

Table 3-12. Devices that have Access to the Detail Screen

Device Name Displayed	Device Description
MODEL B	MODEL B unit setting
90-30 PLC	Number of ports setting
I/O adptr	Number of ports setting
JEMA PC	Number of ports setting
R-J2 Mate	Number of ports setting
Unknown	Number of ports setting

You can add a comment for every device. Comment data is linked to rack, slot and device type. After hardware configuration is changed, if rack, slot and device type are matched, the comment of this device is displayed. If rack, slot or device type are not matched, the comment of this device is not displayed.

 CAUTION

CLR_ASG clears assignments of all ports on all units, including process I/O, model A, model B, and PLC devices. The next time the controller is turned on, ports for these devices will be given default assignments.

3.6.2 Model B I/O Detail Information

Use Procedure 3-15 to access detail information for Model B I/O.

Procedure 3-15 Accessing the Model B I/O Detail Screen

Condition ■ You are already at the I/O Link Device screen.

- Step**
- 1 Move the cursor to Model B.
 - 2 Press F3, DETAIL. You will see a screen similar to the following.

I/O Link Device		JOINT	100%
Model B		Rack 1	1/30
Slot Base	Exp.	Comment	
1 *****	*****	[]
2 *****	*****	[]
3 *****	*****	[]
30 *****	*****	[]
[TYPE]	LIST	[CHOICE]	CLR_ASG

- 3 To list valid base unit product names,
 - a Move the cursor to Base.
 - b Press F4, [CHOICE].

1 *****	5 BOA12A1		
2 BID16A1	6 BIA16P1		
3 BOD16A1	7 BMD88Q1		
4 BMD88A1	8		
Slot Base	Exp.	Comment	
1 *****	*****	[
2 *****	*****	[
3 *****	*****	[
30 *****	*****	[
[TYPE]	LIST	[CHOICE]	CLR_ASG

4 Select the appropriate base unit name.

NOTE “*****” choice indicates no unit.

You will see a screen similar to the following.

I/O Link Device		JOINT	100%
Model B		Rack 1	1 30
Slot Base	Exp.	Comment	
1	B0D16A1 *****	[]
2	***** *****	[]
3	***** *****	[]
30	***** *****	[]
[TYPE]	LIST	[CHOICE]	CLR_ASG

5 To list valid expansion units (which have a “P” in the Product Name),

- a Move the cursor to Exp.
- b Press F4, [CHOICE]. If the base column is not filled in, the message “No base unit” is displayed.

6 To enter a comment, move the cursor to Comment and press ENTER. The teach pendant editor is invoked so you can enter a comment. Information usually entered here involves the mounting location or purpose of the unit.

NOTE If you press FCTN and then select SAVE from any I/O screen, all configuration data is saved, with this setup data (including the comments) into DIOCFGSV.IO.

7 If you save DIOCFGSV.IO from a model A I/O menu or from the FILE [BACKUP] menu, you must also save the Model B I/O Setup data and comments.

CAUTION

CLR_ASG clears assignments of all ports on all units, including process I/O, model A, model B, and PLC devices. The next time the controller is turned on, ports for these devices will be given default assignments.

8 If you press F5 (CLR_ASG), the following message is displayed.
Clear all assignments?

- Press F4, YES to clear all I/O assignments.
- Press F5, NO not to clear all I/O assignments.

9 After setting up detail information, you must power down the controller. Then power it back up for the new information to take effect.

3.6.3 Setting Number of Ports

When the cursor is on the line of 90-30 PLC, I/O Connect, JEMA PC, R-J Mate, R-J2 Mate, or Unknown in the I/O link device screen, press F3, DETAIL. You will see a screen similar to the following.

I/O Link Device		JOINT	100%
90-30 PLC		Rack 1	Slot 1
Port Name		Points	
1 Digital Input		0	
2 Digital Output		0	
[TYPE]	LIST	CLR_ASG	

CAUTION

CLR_ASG clears assignments of all ports on all units, including process I/O, model A, model B, and PLC devices. The next time the controller is turned on, ports for these devices will be given default assignments.

Specify the number of ports needed for your device.

NOTE After setting up number of port information, you must turn off the controller. Then turn it back on for the new information to take effect.

3.7 CONTROLLING I/O

Controlling I/O allows you to test the I/O in your system for proper function during testing operations. Controlling I/O includes:

- Forcing outputs
- Simulating inputs and outputs

3.7.1 Forcing Outputs

Forcing outputs is turning output signals on or off. Outputs can also be forced within a program using I/O instructions. Refer to Section 6.9 in this manual, or to the *FANUC Robotics SYSTEM R-J2 Controller KAREL Reference Manual*. Use Procedure 3-16 to force outputs outside of a program.

Procedure 3-16 Forcing Outputs

- 1 Press F1, [TYPE].
- 2 Select the kind of output you want to force: digital, analog, group, robot, UOP, or SOP.



WARNING

Forcing digital outputs causes connected devices to function. Make certain you know what the digital output is connected to and how it will function before forcing it; otherwise, you could injure personnel or damage equipment.

For digital outputs for example, you will see a screen similar to the following.

I/O Digital Out				WORLD	10%
#	SIM	STATUS			
DO[1]	U	OFF	[]
DO[2]	U	ON	[]
DO[3]	U	OFF	[]
DO[4]	U	OFF	[]
DO[5]	U	OFF	[]
DO[6]	U	ON	[]
DO[7]	U	OFF	[]
DO[8]	U	OFF	[]
DO[9]	U	OFF	[]
DO[10]	U	OFF	[]
[TYPE]	CONFIG	IN/OUT	ON		OFF

DO[4] U **OFF**

- 3 Move the cursor to the STATUS of the output you want to force.

- 4 Press the function key that corresponds to the value you want.

For digital, robot, UOP, and SOP outputs, press

- F4 for ON
- F5 for OFF

```
AO[ 4] U 12H
```

For analog and group outputs, move the cursor to value, and use the numeric keys to type the value. Value entry is always in decimal format. To change the displayed value from decimal to hexadecimal, press F4, FORMAT. Hexadecimal numbers are followed by an “H” on the screen.

3.7.2 Simulating Inputs and Outputs

Simulating inputs and outputs is forcing inputs and outputs without signals entering or leaving the controller. Simulate I/O to test program logic and motion when I/O devices and signals are not set up. You can simulate digital, analog, and group I/O only; you cannot simulate robot, UOP, or SOP I/O. When you are finished simulating a signal you can reset, or *unsimulate*, it. Use Procedure 3-17 to simulate and unsimulate I/O.

Procedure 3-17 Simulating and Unsimulating Inputs and Outputs

- Condition** ■ The input or output has been configured. Refer to Section 3.1.

- Step**
- 1 Press MENUS.
 - 2 Select I/O.
 - 3 Press F1, [TYPE].
 - 4 Select the type of input or output you want to simulate: digital, analog, or group.

For digital inputs for example, you will see a screen similar to the following.

I/O Digital Input				WORLD	10%
#	SIM	STATUS			
DI[1]	U	OFF	[]
DI[2]	S	ON	[]
DI[3]	U	OFF	[]
DI[4]	U	OFF	[]
DI[5]	U	OFF	[]
DI[6]	U	ON	[]
DI[7]	U	OFF	[]
DI[8]	S	OFF	[]
DI[9]	U	OFF	[]
DI[10]	U	OFF	[]

[TYPE] CONFIG IN/OUT SIMULATE UNSIM

DO[4]	OFF
-----	----	-----

- 5 **If you simulate a signal**, you can force the status by setting it to a value. When the signal is unsimulated, its actual status is displayed.
- 6 Move the cursor to the SIM column of the signal you want to simulate.
 - **U** means the signal is not simulated or unsimulated.
 - **S** means the signal is simulated.
- 7 Simulate or unsimulate the signal.
 - **To simulate**, press F4, SIMULATE.
 - **To unsimulate**, press F5, UNSIM.
- 8 **To unsimulate all simulated signals**, press FCTN and then select UNSIM ALL I/O.

NOTE If you disable Digital/Analog I/O from the TEST CYCLE Setup screen, I/O might appear to be simulated when it actually is not. For simulation to occur, you must enable I/O on the TEST CYCLE Setup screen.

3.8 FRAMES SETUP

A frame is a set of three planes at right angles to each other. The point where all three planes intersect is the origin of the frame. This set of planes is called a Cartesian coordinate system. In the robot system, the intersecting edges of the planes are the x, y, and z axes of the frame.

How Frames are Used

Frames are used to describe the location and orientation of a position. The location is the distance in the x, y, and z directions from the origin of the reference frame. The orientation is the rotation about the x, y, and z axes of the reference frame. When you record a position, its location and orientation are automatically recorded as x, y, z, w, p, and r relative to the origin of the frame it uses as a reference.

The location of a position is expressed in millimeters as three dimensions. For example, 300,425,25 means the position is 300mm in the x direction, 425mm in the y direction, and 25mm in the z direction from the origin.

The orientation of a position is expressed in degrees as three dimensions. For example, 0,-90,0 means that the position is rotated -90 degrees about the y axis and is not rotated about the x or z axes.

Kinds of Frames

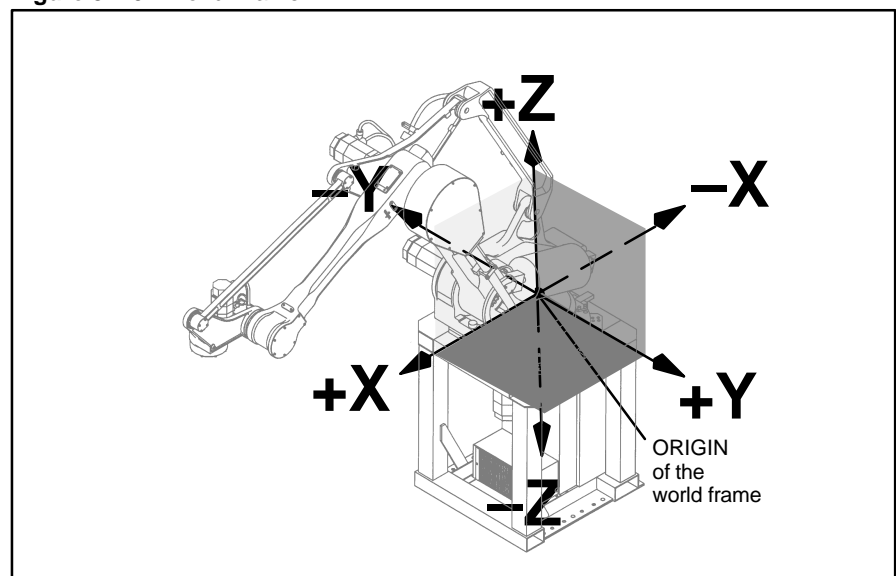
The robot system uses four kinds of frames. The different kinds of frames make it easier to do certain tasks. The four kinds of frames are

- World frame – the default frame of the robot
- Tool frame – a user-defined frame
- User frame – a user-defined frame
- Jog frame – a user-defined frame

World Frame

The world frame is a default frame that cannot be changed. The origin of the world frame (0,0,0,0,0,0) is the reference position for user frame and jog frame. The origin is located at a predefined position within the robot. See Figure 3-20.

Figure 3-20. World Frame



Tool Frame

The tool frame is a Cartesian coordinate system that has the position of the tool center point (TCP) at its origin. You must set the tool frame to define the point on the gripper at which the palletizing is to be done.

User Frame

The user frame is the reference frame for all recorded positional data in a program. You can define this frame anywhere.

Remote TCP (RTCP) Frame

The remote TCP (RTCP) frame is a kind of user frame you must define in order to use remote TCP jogging and the remote TCP motion option. You define this frame using the location of the remote TCP as the origin of the frame.

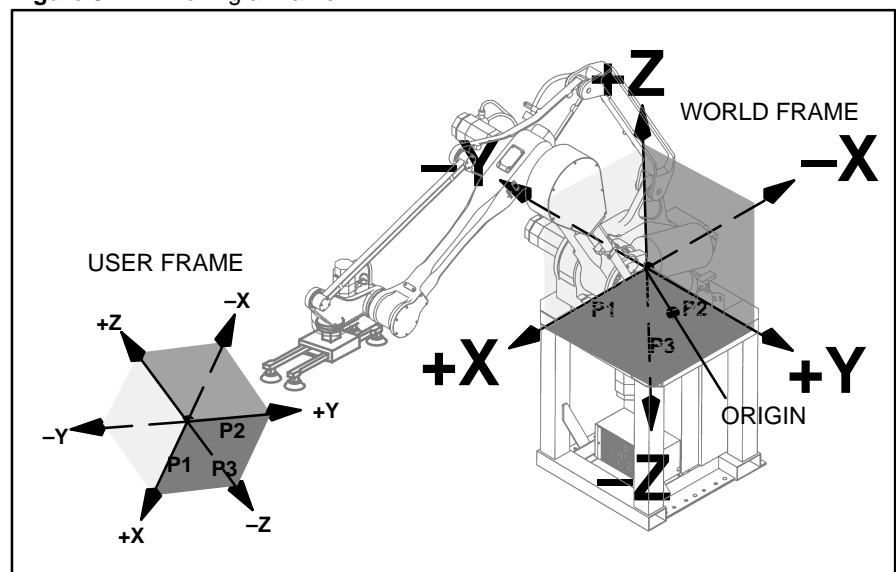
Jog Frame

The jog frame is a frame in which to jog easily.

Moving the Location and Orientation of a Frame

You can move the location and orientation of any frame except the world frame. When you move the location or orientation of a frame, all positions recorded with that frame also move. However, the location of those positions will stay the same within that frame. See Figure 3-21.

Figure 3-21. Moving a Frame



CAUTION

If you change any **TOOL** or **USER** frame data after a program has been taught, you must reset each program position or range. If you do not, **damage could occur** to the equipment.

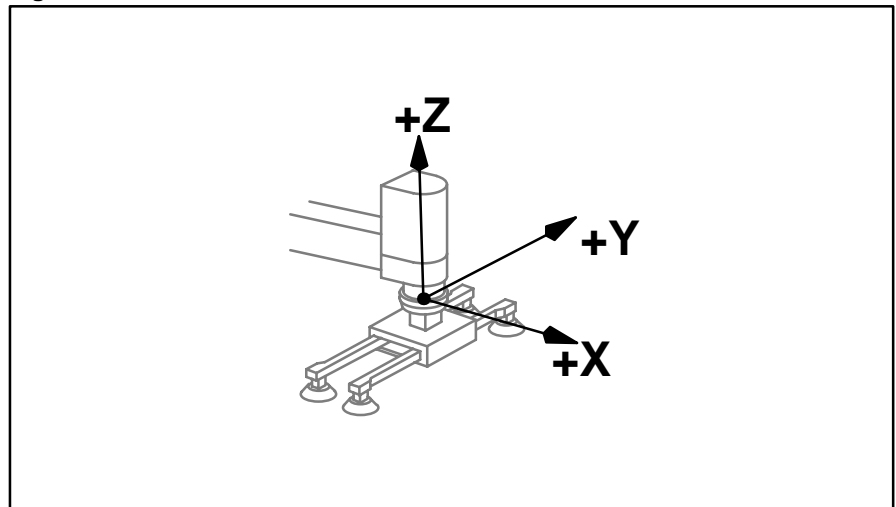
3.8.1 Setting Up Tool Frame

By default, the origin of the tool frame is on the faceplate of the robot. You must move the origin of the tool frame to the position, both location and orientation, where the work is to be done. This position is called the tool center point (TCP). See Figure 3-22.

All measurements in tool frame are relative to the origin of tool frame.

- Before you use tool frame, you must set up its location and orientation.
- You can set up as many as six different tool frames for each robot. They will be stored in the system variable \$MNUTOOL.
- You can select one tool frame to be active. The frame number will be stored in system variable \$MNUTOOLNUM.
- You can jog the robot in tool frame.

Figure 3-22. Tool Frame



You can use three methods to define the tool frame:

- Three point method
- Six point method
- Direct entry method

NOTE If you have a 4-axis robot (such as an A-520i, M-400i, or M-410i), you can define a tool frame using only the direct entry method.

Three Point Method

Use the three point method to define the **location** of the tool frame when the values cannot be measured and directly entered. The three approach points must be taught with the tool touching a common point from three different approach directions.

Use Procedure 3-18 to set up the tool frame using the three point method.

Six Point Method

Use the six point method to define the **location and orientation** of the tool frame when the values cannot be measured and directly entered. The six point method requires three points that define the direction vector for the tool, and the three points that define the location of the tool center point.

Use Procedure 3-19 to set up the tool frame using the six point method.

Direct Entry Method

The direct entry method provides direct recording and numerical entry of the frame position. For TCP dimensions, refer to the manufacturing specifications of the tool. Use Procedure 3-20 to set up the tool frame using the direct entry method.

Use Procedure 3-21 to select a tool frame.

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

**WARNING**

If you are setting up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.

If you are modifying an existing frame, make sure that all frame data is set the way you want before you change it.

Otherwise, you could injure personnel or damage equipment.

NOTE If you have a 4-axis robot (such as an A-520i, M-400i, or M-410i), you can define a tool frame using only the direct entry method.

- Step**
- 1 Press MENU.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select Frames.
 - 5 **To choose the motion group** for the frame you are setting up in systems with multiple motion groups, press F3, [OTHER], and select the group you want: Group 1, Group 2, or Group 3. The default motion group is group 1.
 - 6 **If tool frames are not displayed**, press F3, [OTHER], and select Tool Frame. If F3, [OTHER], is not displayed, press PREV.
 - 7 **To display the settings for all frames**, press PREV repeatedly until you see a screen similar to the following.

SETUP Frames				JOINT	50%
Tool Frame Setup / Three Point				1/6	
	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	*****	
ACTIVE TOOL \$MNUTOOLNUM[1]=1					
[TYPE] DETAIL [OTHER] CLEAR SETIND					

- 8 **To set the numerical values to zero**, move the cursor to the frame number and press F4, CLEAR.
- 9 Press F2, DETAIL.

Input frame number to display:

- 10 To select a frame,
 - a Press F3, FRAME.
 - b Type the desired frame number.
 - c Press ENTER.
- 11 Press F2, [METHOD].
- 12 Select Three Point. You will see a screen similar to the following.

```

SETUP Frames                                JOINT 50%
Tool Frame Setup / Three Point                1/4
Frame number: 2
X: 0.0      Y: 0.0      Z: 0.0
W: 0.0      P: 0.0      R: 0.0
Comment: *****
Approach point 1:      UNINIT
Approach point 2:      UNINIT
Approach point 3:      UNINIT

Active TOOL $MNUTOOLNUM[1]=1
[ TYPE ] [METHOD] FRAME
  
```

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

NOTE Record the three approach points with the tool tip touching the same point from three different approach directions.

Approach point 1: UNINIT

- 14 Record the first approach point:
 - a Move the cursor to Approach point 1.
 - b Jog the robot so that the tool tip touches a reference point.
 - c Press and hold the SHIFT key and press F5, RECORD.

Approach point 2: UNINIT

- 15 Record the second approach point:
 - a Move the cursor to Approach point 2.
 - b Rotate the faceplate at least 90° (but no more than 360°) about the z axis of the tool coordinates.
 - c Jog the robot so that the tool tip touches the reference point used in Step 14.
 - d Press and hold the SHIFT key and press F5, RECORD.

Approach point 3:

UNINIT

- 16** Record the third approach point:
 - a** Move the cursor to Approach point 3.
 - b** Rotate the tool about either the x or y axis of the tool coordinates.
 - c** Jog the robot so that the tool tip touches the reference point used in Step 14.
 - d** Press and hold the SHIFT key and press F5, RECORD.
- 17** **To move to a recorded position**, move the cursor to the desired position, press and hold the SHIFT key and press F4, MOVE_TO.
- 18** **To select the tool frame to use**, press F5, SETIND, type the desired frame number and press ENTER.

NOTE To select the number of the tool frame you want to use, you can also use the jog menu. Refer to Section 2.2.8.

 **CAUTION**

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving Frame Configuration

- 19** **To save the frames and related system variables to a file**,
 - a** Press MENU.
 - b** Select FILE.
 - c** Press F1, [TYPE].
 - d** Select File.
 - e** Press F5, [UTIL].
 - f** Select Set Device.
 - g** Move the cursor to the device you want and press ENTER.
 - h** Press FCTN.
 - i** Select SAVE. This will save the frame positions and comments for all frames to the file, FRAMEVAR.SV, on the default device.
- Display the SYSTEM Variables menu,**
- j** Press MENU.
 - k** Select SYSTEM.
 - l** Press F1, [TYPE].
 - m** Select Variables.
 - n** Press FCTN.
 - o** Select SAVE. The tool frame positions and system variables are saved in the SYSVAR.SV file, on the default device.

Procedure 3-19 Setting Up Tool Frame Using the Six Point Method

**WARNING**

If you are setting up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.

If you are modifying an existing frame, make sure that all frame data is set the way you want before you change it.

Otherwise, you could injure personnel or damage equipment.

NOTE If you have a 4-axis robot (such as an A-520i, M-400i, or M-410i), you can define a tool frame using only the direct entry method.

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select Frames.
 - 5 **To choose the motion group** for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want: Group 1, Group 2, or Group 3. The default motion group is group 1.
 - 6 **If tool frames are not displayed**, press F3, [OTHER], and select Tool Frame. If F3, [OTHER], is not displayed, press PREV.
 - 7 **To display the settings for all frames**, press PREV repeatedly until you see a screen similar to the following.

SETUP Frames				JOINT	50%
Tool Frame Setup / Six Point				1/6	
	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	*****	
ACTIVE TOOL \$MNUTOOLNUM[1]=1					
[TYPE] DETAIL [OTHER] CLEAR SETIND					

- 8 **To set the numerical values to zero**, move the cursor to the frame number and press F4, CLEAR.
- 9 Press F2, DETAIL.

Input frame number to display:

- 10 To select a frame,
 - a Press F3, FRAME.
 - b Type the desired frame number.
 - c Press ENTER.
- 11 Press F2, [METHOD].
- 12 Select Six Point. You will see a screen similar to the following.

```

SETUP Frames JOINT 50%
Tool Frame Setup/ Six Point 1/7
Frame number: 2
  X:  0.0    Y:  0.0    Z:  0.0
  W:  0.0    P:  0.0    R:  0.0
Comment: *****
Approach point 1:    UNINIT
Approach point 2:    UNINIT
Approach point 3:    UNINIT
Orient Origin Point: UNINIT
X Direction Point:  UNINIT
Z Direction Point:  UNINIT
ACTIVE TOOL $MNUTOOLNUM[1]=1
[ TYPE ] [METHOD] FRAME MOVE TO RECORD
  
```

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

NOTE Record the three approach points with the tool tip touching the same point from three different approach directions.

Approach point 1: UNINIT

- 14 Record the first approach point:
 - a Move the cursor to Approach point 1.
 - b Jog the robot so that the tool tip touches a reference point.
 - c Press and hold the SHIFT key and press F5, RECORD.

Approach point 2:

UNINIT

15 Record the second approach point:

- a Move the cursor to Approach point 2.
- b Rotate the faceplate at least 90° (but no more than 360°) about the z axis of the tool coordinates.
- c Jog the robot so that the tool tip touches the reference point used in Step 14.
- d Press and hold the SHIFT key and press F5, RECORD.

Approach point 3:

UNINIT

16 Record the third approach point:

- a Move the cursor to Approach point 3.
- b Rotate the tool about either the x or y axis of the tool coordinates.
- c Jog the robot so that the tool tip touches the reference point used in Step 14.
- d Press and hold the SHIFT key and press F5, RECORD.

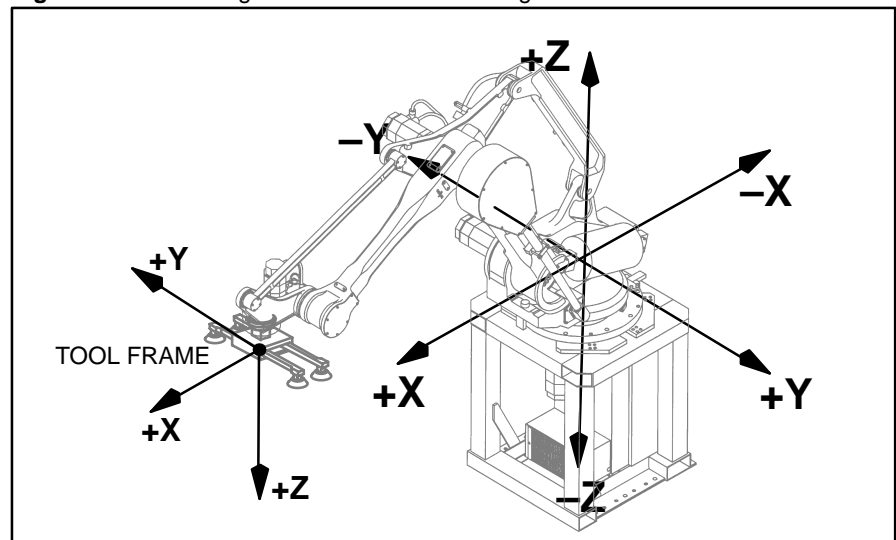
17 Define the orientation of the origin:

Orient Origin Point:

UNINIT

- a Move the cursor to Orient Origin Point.
- b Jog the robot so that the pointer is parallel to the z axis of the world frame, pointing in the $-z$ direction. Make sure that the x axis of the tool is parallel to the x axis of the world frame. See Figure 3-23.
- c Press and hold the SHIFT key and press F5, RECORD.

Figure 3-23. Defining the Orientation of the Origin



18 Define the +X Direction Point:**X Direction Point:** UNINIT

- a** Move the cursor to X Direction Point.
- b** Change the jog coordinate system to WORLD.
- c** Jog the robot so that the tool moves in the +x direction. For example, if the x axis of the tool is aligned with the world x axis, jog in the +x direction.

NOTE To assist you in moving the tool in the +x direction, move the tool at least 250mm or more.

- d** Press and hold the SHIFT key and press F5, RECORD.

19 Define the +Z Direction Point:**Z Direction Point:** UNINIT

- a** Move the cursor to Orient Origin Point.
- b** Press and hold the SHIFT key and press F4, MOVE_TO, to move the robot to the Orient Origin Point.
- c** Move the cursor to Z Direction Point.
- d** Jog the robot in the -z direction.
- e** Press and hold the SHIFT key and press F5, RECORD.

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

21 **To select the tool frame to use**, press F5, SETIND, type the desired frame number and press ENTER.

NOTE To select the number of the tool frame you want to use, you can also use jog menu. Refer to Section 2.2.8.

⚠ CAUTION

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving Frame Configuration**22 To save the frames and related system variables to a file.**

- a** Press **MENU**.
- b** Select **FILE**.
- c** Press **F1**, [**TYPE**].
- d** Select **File**.
- e** Press **F5**, [**UTIL**].
- f** Select **Set Device**.
- g** Move the cursor to the device you want and press **ENTER**.
- h** Press **FCTN**.
- i** Select **SAVE**. This will save the frame positions and comments for all frames to the file, **FRAMEVAR.SV**, on the default device.

Display the SYSTEM Variables menu,

- j** Press **MENU**.
- k** Select **SYSTEM**.
- l** Press **F1**, [**TYPE**].
- m** Select **Variables**.
- n** Press **FCTN**.
- o** Select **SAVE**. The tool frame positions and system variables are saved in the **SYSVAR.SV** file, on the default device.

Procedure 3-20 Setting Up Tool Frame Using the Direct Entry Method

**WARNING**

If you are setting up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.

If you are modifying an existing frame, make sure that all frame data is set the way you want before you change it.

Otherwise, you could injure personnel or damage equipment.

NOTE If you have a 4-axis robot (such as an A-520i, M-400i, or M-410i), you can define a tool frame using only the direct entry method.

- Step**
- 1 Press MENU.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select Frames.
 - 5 **To choose the motion group** for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want: Group 1, Group 2, or Group 3. The default motion group is group 1.
 - 6 **If tool frames are not displayed**, press F3, [OTHER], and select Tool Frame. If F3, [OTHER], is not displayed, press PREV.
 - 7 **To display the settings for all the frames**, press PREV repeatedly until you see a screen similar to the following.

SETUP Frames				JOINT	50%
Tool Frame Setup / Direct Entry				1/6	
	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	*****	
Active TOOL \$MNUTOOLNUM[1]=1					
[TYPE] [DETAIL] [OTHER] [CLEAR] [SETIND]					

- 8 **To set the numerical values to zero**, move the cursor to the frame number and press F4, CLEAR.
- 9 Press F2, DETAIL.

Input frame number to display:

- 10 To select a frame,
 - a Press F3, FRAME.
 - b Type the desired frame number.
 - c Press ENTER.
- 11 Press F2, [METHOD].
- 12 Select Direct Entry. You will see a screen similar to the following.

```

SETUP Frames                               JOINT 50%
Tool Frame Setup / Direct Entry           1/7
Frame number: 1
1 Comment: *****
2 X:                                       0.000
3 Y:                                       0.000
4 Z:                                       0.000
5 W:                                       0.000
6 P:                                       0.000
7 R:                                       0.000
Configuration:   N R D B, 0, 0, 0

Active TOOL $MNUTOOLNUM[1]=1
[ TYPE ] [METHOD] FRAME

```

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

```

2 X 0.000
3 Y: 0.000
4 Z: 0.000
5 W: 0.000
6 P: 0.000
7 R: 0.000
Configuration: N R D B,0,0,0

```

- 14 Set each position component:
 - a Move the cursor to the component.
 - b Enter the numeric value for the component.
 - c Press the ENTER key to set the new value.
- 15 To select the tool frame to use, press F5, SETIND, type the desired frame number and press ENTER.

NOTE To select the number of the tool frame you want to use, you can also use jog menu. Refer to Section 2.2.8.

⚠ CAUTION

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving Frame Configuration**16 To save the frames and related system variables to a file,**

- a** Press **MENUS**.
- b** Select **FILE**.
- c** Press **F1**, [**TYPE**].
- d** Select **File**.
- e** Press **F5**, [**UTIL**].
- f** Select **Set Device**.
- g** Move the cursor to the device you want and press **ENTER**.
- h** Press **FCTN**.
- i** Select **SAVE**. This will save the frame positions and comments for all frames to the file, **FRAMEVAR.SV**, on the default device.

Display the SYSTEM Variables menu,

- j** Press **MENUS**.
- k** Select **SYSTEM**.
- l** Press **F1**, [**TYPE**].
- m** Select **Variables**.
- n** Press **FCTN**.
- o** Select **SAVE**. The tool frame positions and system variables are saved in the **SYSVAR.SV** file, on the default device.

Procedure 3-21 Selecting a Tool Frame

Condition
Step

NOTE To select the number of tool frame you want to use, you can also use jog menu. Refer to Section 2.2.8.

- The tool frame you want to select has been set up.

- 1 Press MENUS.
- 2 Select SETUP.
- 3 Press F1, [TYPE].
- 4 Select Frames.
- 5 **To choose the motion group** for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want: Group 1, Group 2, or Group 3. The default motion group is group 1.
- 6 **If tool frames are not displayed**, press F3, [OTHER], and select Tool Frame. If F3, [OTHER], is not displayed, press PREV. You will see a screen similar to the following.

SETUP Frames				JOINT 50%
Tool Frame	Setup /	Direct	Entry	1/6
	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	*****
Active TOOL \$MNUTOOLNUM[1]=1				
[TYPE]	DETAIL	[OTHER]	CLEAR	SETIND

- 7 **To select the tool frame to use**, press F5, SETIND, type the desired frame number and press ENTER.

NOTE To select the number of the tool frame you want to use, you can also use jog menu. Refer to Section 2.2.8.

- 8 When a position is recorded in the teach pendant program, the value of the position's tool frame will always equal the value of \$MNUTOOLNUM[group_no] at the time the position was recorded. When a teach pendant program is executed, you must make sure that the tool frame of the position equals the value of \$MNUTOOLNUM [group_no], otherwise, an error will occur. Set the value of \$MNUTOOLNUM using the UTOOL_NUM=n instruction in the teach pendant program before you record the position to guarantee that the tool frame numbers match during program execution. Refer to Section 6.14 for more information on the UTOOL_NUM instruction.

3.8.2 Setting Up User Frame

User frame is a frame that you can set up in any location, with any orientation.



CAUTION

Recorded positions are not affected by UFRAME and UFRAME has no affect during playback. However, **position registers are recorded with respect to UFRAME**. If you change UFRAME, any recorded position registers will also change.

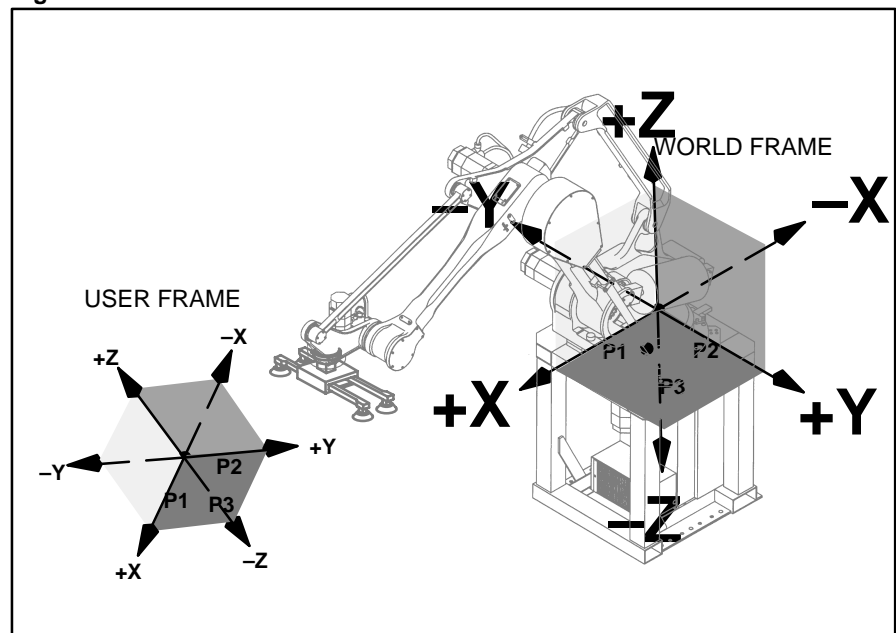
User frames are used so that position registers in a program can be recorded relative to the origin of the frame. All position registers in a program are automatically recorded in user frame. If you do not set up the location and orientation of the user frame, the user frame will be set by default to the world frame.

After you set up the user frame, you can change its location and orientation. All position registers in a program recorded relative to that frame change with it.

- You can set up as many as six user frames for each robot. They will be stored in the system variable \$MNUFRAME.
- You can select one user frame to be active at a time. The frame number will be stored in \$MNUFRAMENUM.
- You can jog the robot in user frame.

See Figure 3-24.

Figure 3-24. World and User Frames



You can use three methods to define the user frame:

- Three point method
- Four point method
- Direct entry method

Three Point Method

Recording three points defines the user frame. The three points are the origin, a position along the +x-axis of the user frame, and a position on the x-y plane of the user frame (defines the x-y plane and the y-z plane). Use Procedure 3-22 to define the user frame using the three point method.

Four Point Method

Use the four point method when you need to define a frame that has its origin at a position other than the reference of the frame or to define multiple frames with parallel axes. The four points are the reference of the frame (called orient origin point), a point along the +x-axis of the frame (defines the x-z plane), a point on the x-y plane of the frame (defines the x-y plane and the y-z plane) and the origin of the frame (called system origin). Use Procedure 3-23 to define a user frame using the four point method.

Direct Entry Method

Use the direct entry method when you know the coordinates of the user frame. The direct entry method allows you to designate the origin with values for x, y, z, w, p, and r. Use Procedure 3-24 to define a user frame using the direct entry method.

Use Procedure 3-25 to select a user frame.

Procedure 3-22 Setting Up the User Frame Using the Three Point Method



WARNING

If you are setting up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.

If you are modifying an existing frame, make sure that all frame data is set the way you want before you change it.

Otherwise, you could injure personnel or damage equipment.

- | | |
|-------------|---|
| Step | <ol style="list-style-type: none">1 Press MENUS.2 Select SETUP.3 Press F1 [TYPE].4 Select Frames.5 To choose the motion group for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want: Group 1, Group 2, or Group 3. The default motion group is group 1. |
|-------------|---|

- 6** If user frames are not displayed, press F3, [OTHER], and select User Frame. If F3, [OTHER], is not displayed, press PREV.
- 7** To display the settings for all frames, press PREV repeatedly until you see a screen similar to the following.

SETUP Frames				JOINT 50%
User/RTCP Setup / Three Point				1/6
	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	*****
Active UFRAME/RTCP \$MNUFRAMNUM[1]=0				
[TYPE] DETAIL [OTHER] CLEAR SETIND >				
[TYPE] CLRIND >				

- 8** To set the numerical values to zero, move the cursor to the frame number and press F4, CLEAR.
- 9** Press F2, DETAIL.
- 10** To select a frame,
- Press F3, FRAME.
 - Type the desired frame number.
 - Press ENTER.
- 11** Press F2, [METHOD].
- 12** Select Three Point. You will see a screen similar to the following.

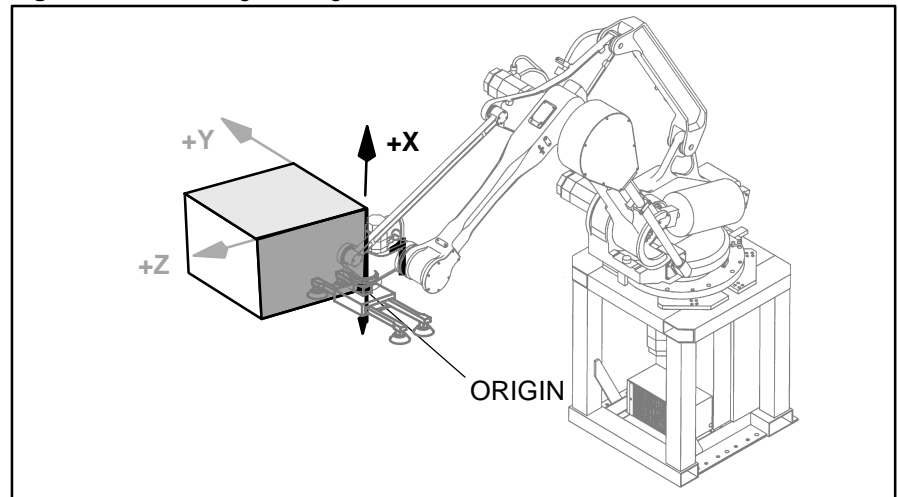
SETUP Frames				JOINT 50%
User/RTCP Frame Setup/ Three Point				1/4
Frame number: 2				
X:	0.0	Y:	0.0	Z: 0.0
W:	0.0	P:	0.0	R: 0.0
Comment:	*****			
Orient Origin Point:	UNINIT			
X Direction Point:	UNINIT			
Y Direction Point:	UNINIT			
Active UFRAME \$MNUFRAMNUM[1]=0				
[TYPE] [METHOD] FRAME				

- 13** To add a comment:
- Move the cursor to the comment line and press ENTER.
 - Select a method of naming the comment.
 - Press the appropriate function keys to enter the comment.
 - When you are finished, press ENTER.

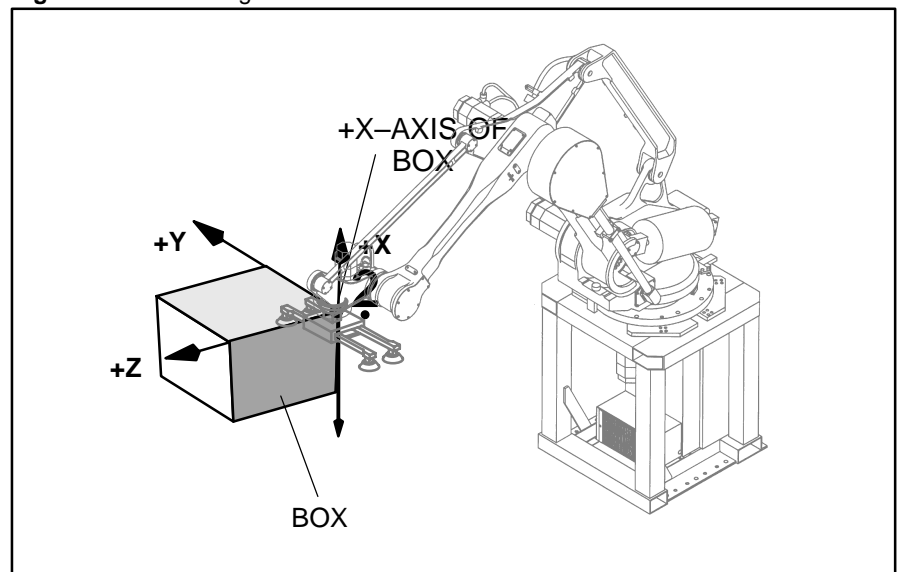
Input frame number to display:

14 Define the origin point of the user frame.**Orient Origin Point:** UNINIT

- a** Move the cursor to Orient Origin Point.
- b** Jog the robot TCP to the origin. In Figure 3-25, the origin is labeled 1.
- c** Press and hold the SHIFT key and press F5, RECORD.

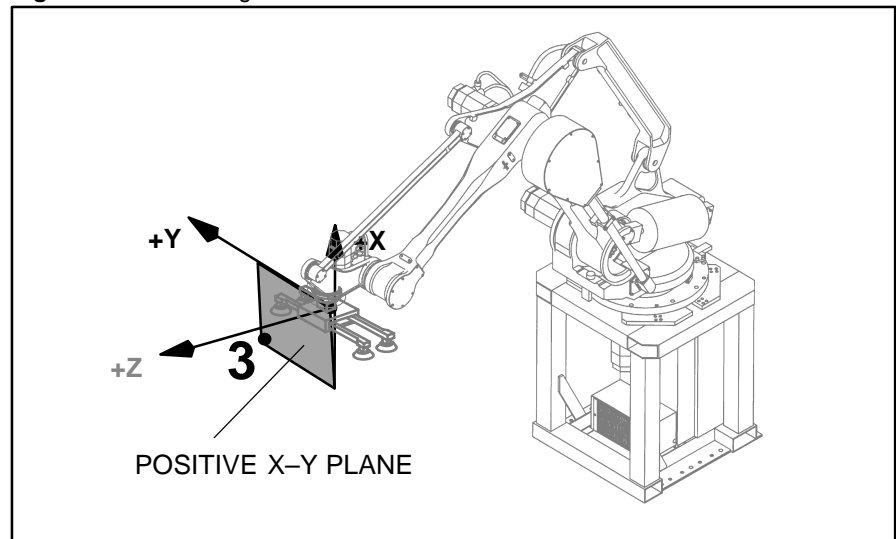
Figure 3-25. Defining the Origin**15** Define the +X Direction Point:**X Direction Point:** UNINIT

- a** Move the cursor to X Direction Point.
- b** Jog the robot to a point along the x-axis of the box. In Figure 3-26, this point is labeled 2.
- c** Press and hold the SHIFT key and press F5, RECORD.

Figure 3-26. Defining the X Direction Point

16 Define a point on the positive X-Y plane:Y Direction Point: **UNINIT**

- a Move the cursor to Y Direction Point.
- b Jog the robot to a location on the positive X-Y plane. In Figure 3-27, this point is labeled number 3.
- c Press and hold the SHIFT key and press F5, RECORD.

Figure 3-27. Defining the X-Y Plane

17 To move to a recorded position, move the cursor to the desired position, press and hold the SHIFT key and press F4, MOVE_TO.

18 To select the user frame to use, press F5, SETIND, type the frame number and press ENTER. This sets the active user frame (\$MNUFRAMNUM[1]) to the number of the frame you want.

NOTE To select the number of the user frame you want to use, you can also use the jog menu. Refer to Section 2.2.8.

19 To clear the current user frame select, press NEXT, >, and then F2, CLRIND. This sets the active user frame (\$MNUFRAMNUM[1]) to zero, which means that the default user frame is currently selected.

⚠ CAUTION

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving Frame Configuration

- 20 To save the frames and related system variables to a file** on the default device,
- a** Press **MENUS**.
 - b** Select **FILE**.
 - c** Press **F1**, [**TYPE**].
 - d** Select **File**.
 - e** Press **F5**, [**UTIL**].
 - f** Select **Set Device**.
 - g** Move the cursor to the device you want and press **ENTER**.
 - h** Press **FCTN**.
 - i** Select **SAVE**. This will save the frame positions and comments for all frames to the file, **FRAMEVAR.SV**, on the default device.
- Display the SYSTEM Variables menu,**
- j** Press **MENUS**.
 - k** Select **SYSTEM**.
 - l** Press **F1**, [**TYPE**].
 - m** Select **Variables**.
 - n** Press **FCTN**.
 - o** Select **SAVE**. The frame positions and system variables are saved in the **SYSVAR.SV** file, on the default device.

Procedure 3-23 Setting Up User Frame Using the Four Point Method

**WARNING**

If you are setting up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.

If you are modifying an existing frame, make sure that all frame data is set the way you want before you change it.

Otherwise, you could injure personnel or damage equipment.

- Condition** ■ The tool frame is set up and selected.
- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select Frames.
 - 5 **To choose the motion group** for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want: Group 1, Group 2, or Group 3. The default motion group is group 1.
 - 6 **If user frames are not displayed**, press F3, [OTHER], and select User Frame. If F3, [OTHER], is not displayed, press PREV.
 - 7 **To display the settings for all frames**, press PREV repeatedly until you see a screen similar to the following.

SETUP Frames				JOINT 50%
User/RTCP Setup/ Four Point				1/6
	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	*****
Active UFRAME/RTCP \$MNUFRAMNUM[1]=1				
[TYPE] DETAIL [OTHER] CLEAR SETIND >				
[TYPE] CLRIND >				

- 8 **To set the numerical values to zero**, move the cursor to the frame number and press F4, CLEAR.
- 9 Press F2, DETAIL.

Input frame number to display:

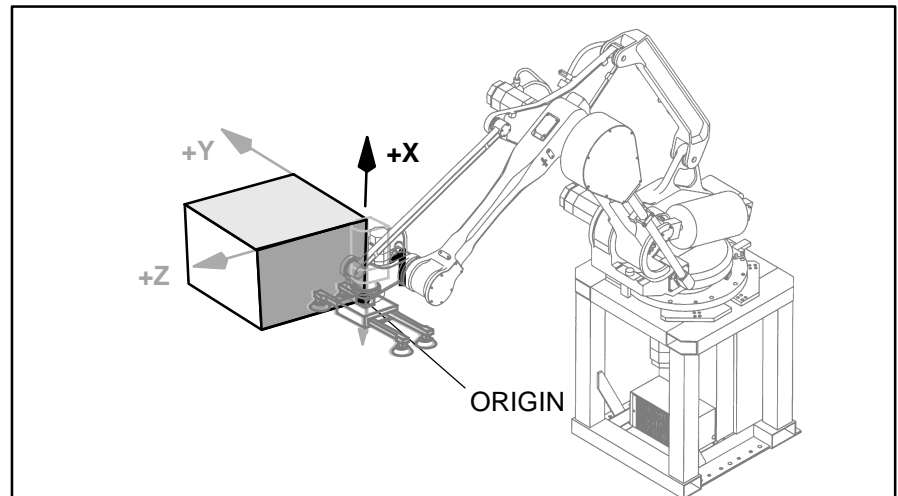
- 10 To select a frame,
 - a Press F3, FRAME.
 - b Type the desired frame number.
 - c Press ENTER.
- 11 Press F2, [METHOD].
- 12 Select Four Point. You will see a screen similar to the following.

SETUP Frames			JOINT	50%	
User/RTCP Frame Setup/ Four Point			1/5		
Frame number: 2					
X:	0.0	Y:	0.0	Z:	0.0
W:	0.0	P:	0.0	R:	0.0
Comment: *****					
Orient Origin Point:			UNINIT		
X Direction Point:			UNINIT		
Y Direction Point:			UNINIT		
System Origin:			UNINIT		
Active UFRAME \$MNUFRAMNUM[1]=0					
[TYPE] [METHOD] FRAME					

- 13 To add a comment:
 - a Move the cursor to the comment line and press ENTER.
 - b Select a method of naming the comment.
 - c Press the appropriate function keys to enter the comment.
 - d When you are finished, press ENTER.
- 14 Define the reference point of the user frame.
 - a Move the cursor to Orient Origin Point.
 - b Jog the robot TCP to the origin. In Figure 3-28, the origin is labeled 1.
 - c Press and hold the SHIFT key and press F5, RECORD.

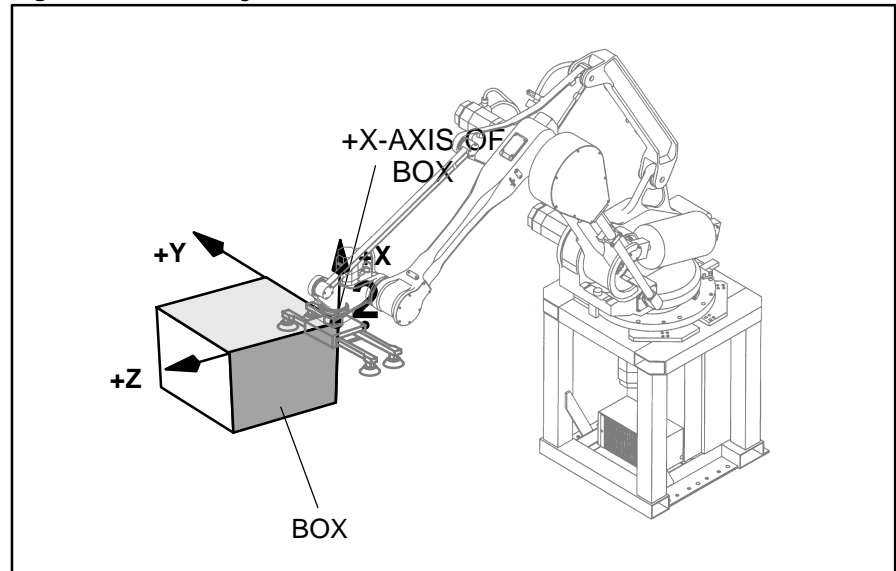
Orient Origin Point: UNINIT

Figure 3-28. Defining the Origin

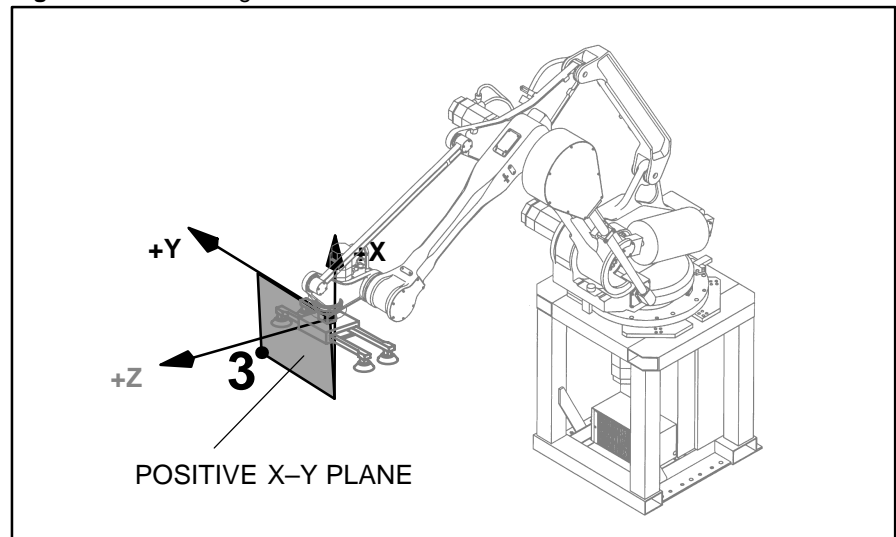


15 Define the +X Direction Point:**X Direction Point:** UNINIT

- a** Move the cursor to X Direction Point.
- b** Jog the robot TCP to a point along the +x-axis of the box. In Figure 3-29, the origin is labeled 2.
- c** Press F5, RECORD, to record a position.

Figure 3-29. Defining the X Direction Point**16** Define a point on the X-Y plane:**Y Direction Point:** UNINIT

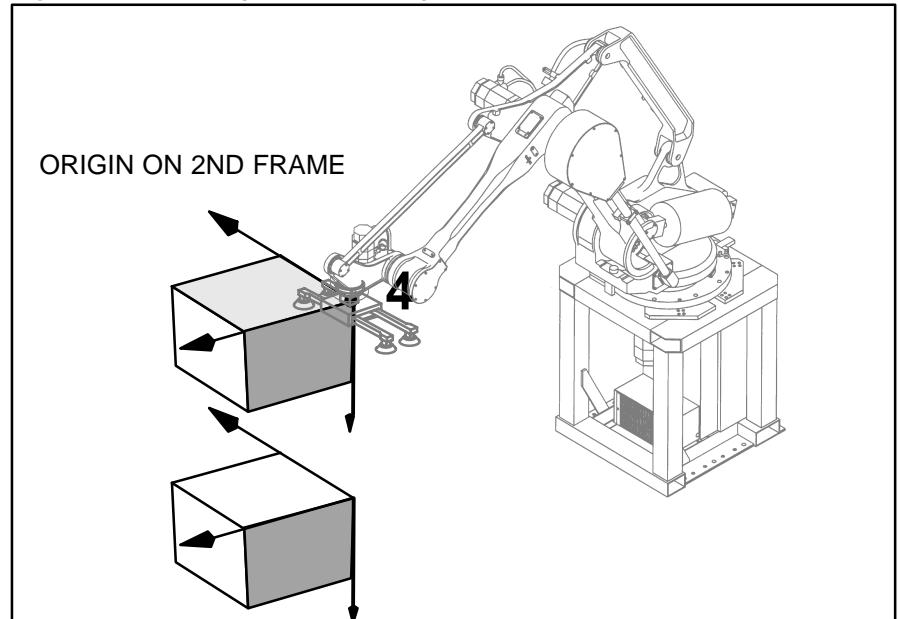
- a** Move the cursor to Y Direction Point.
- b** Jog the robot to a location on the positive X-Y plane. In Figure 3-30, this point is labeled number 3.
- c** Press and hold the SHIFT key and press F5, RECORD.

Figure 3-30. Defining the X-Y Plane

System Origin Point: **UNIT1**

- 17 Teach the origin of the second user frame.
 - a Move the cursor to System Origin Point.
 - b Jog the robot TCP to the origin of the second user frame. In Figure 3-31, the origin is labeled 4.
 - c Press F5, RECORD, to record a position.

Figure 3-31. Defining the Second Origin



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

CAUTION

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

- 19 To set the numerical values to zero, move the cursor to the frame number and press F4, CLEAR.
- 20 To select the user frame to use, press F5, SETIND, type the desired frame number and press ENTER. This sets the active user frame (\$MNUFRAMNUM[1]) to the number of the frame you want.

NOTE To select the number of the user frame you want to use, you can also use the jog menu. Refer to Section 2.2.8.

- 21 To clear the current frame to zero, move the cursor to the frame number and press NEXT, >, and then F2, CLRIND. This sets the active user frame (\$MNUFRAMNUM[1]) to zero, which means that the default user frame is currently selected.

⚠ CAUTION

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving Frame Configuration

- 22 To save the frames and related system variables to a file** on the default device,
- a** Press **MENUS**.
 - b** Select **FILE**.
 - c** Press **F1**, [**TYPE**].
 - d** Select **File**.
 - e** Press **F5**, [**UTIL**].
 - f** Select **Set Device**.
 - g** Move the cursor to the device you want and press **ENTER**.
 - h** Press **FCTN**.
 - i** Select **SAVE**. This will save the frame positions and comments for all frames to the file, **FRAMEVAR.SV**, on the default device.
- Display the SYSTEM Variables menu,**
- j** Press **MENUS**.
 - k** Select **SYSTEM**.
 - l** Press **F1**, [**TYPE**].
 - m** Select **Variables**.
 - n** Press **FCTN**.
 - o** Select **SAVE**. The frame positions and system variables are saved in the **SYSVAR.SV** file, on the default device.

Procedure 3-24 Setting Up User Frame Using the Direct Entry Method

**WARNING**

If you are setting up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.

If you are modifying an existing frame, make sure that all frame data is set the way you want before you change it.

Otherwise, you could injure personnel or damage equipment.

Condition ■ The tool frame is set up and selected.

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select Frames.
 - 5 **To choose the motion group** for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want: Group 1, Group 2, or Group 3. The default motion group is group 1.
 - 6 **If user frames are not displayed**, press F3, [OTHER], and select User Frame. If F3, [OTHER], is not displayed, press PREV.
 - 7 **To display the settings for all the frames**, press PREV repeatedly until you see a screen similar to the following.

SETUP Frames				JOINT 50%
User/RTCP Setup/ Direct Entry				1/6
	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	*****
Active UFRAME/RTCP \$MNUFRAMNUM[1]=1				
[TYPE] DETAIL [OTHER] CLEAR SETIND >				
[TYPE] CLRIND >				

- 8 **To set the numerical values to zero**, move the cursor to the frame number and press F4, CLEAR.

Input frame number to display:

- 9 Press F2, DETAIL.
- 10 To select a frame,
 - a Press F3, FRAME.
 - b Type the desired frame number.
 - c Press ENTER.
- 11 Press F2, [METHOD].
- 12 Select Direct Entry. You will see a screen similar to the following.

```

SETUP Frames                                JOINT 50%
User/RTCP Frame Setup/ Direct Entry          1/7
Frame number: 1
1 Comment: *****
2 X:                                0.000
3 Y:                                0.000
4 Z:                                0.000
5 W:                                0.000
6 P:                                0.000
7 R:                                0.000
Configuration: N R D B, 0, 0, 0
Active UFRAME $MNUFRAMENUM[1]=0
[ TYPE ] [METHOD] FRAME
  
```

- 13 To add a comment:
 - a Move the cursor to the comment line and press the ENTER key.
 - b Select a method of naming the comment.
 - c Press the appropriate function keys to enter the comment.
 - d When you are finished, press ENTER.
- 14 Set each position component:
 - a Move the cursor to the component.
 - b Enter the numeric value for the component.
 - c Press the ENTER key to set the new value.
- 15 To select the user frame to use, press F5, SETIND, type the desired frame number and press ENTER. This sets the active user frame (\$MNUFRAMNUM[1]) to the number of the frame you want.

NOTE To select the number of the user frame you want to use, you can also use the jog menu. Refer to Section 2.2.8.

```

2 X                                0.000
3 Y:                                0.000
4 Z:                                0.000
5 W:                                0.000
6 P:                                0.000
7 R:                                0.000
Configuration: N R D B,0,0,0
  
```

- 16 To clear the current frame to zero, move the cursor to the frame number and press NEXT, >, and then F2, CLRIND. This sets the active user frame (\$MNUFRAMNUM[1]) to zero, which means that the default user frame is currently selected.

⚠ CAUTION

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving Frame Configuration

- 17 To save the frames and related system variables to a file** on the default device,
- a** Press **MENUS**.
 - b** Select **FILE**.
 - c** Press **F1**, [**TYPE**].
 - d** Select **File**.
 - e** Press **F5**, [**UTIL**].
 - f** Select **Set Device**.
 - g** Move the cursor to the device you want and press **ENTER**.
 - h** Press **FCTN**.
 - i** Select **SAVE**. This will save the frame positions and comments for all frames to the file, **FRAMEVAR.SV**, on the default device.
- Display the SYSTEM Variables menu,**
- j** Press **MENUS**.
 - k** Select **SYSTEM**.
 - l** Press **F1**, [**TYPE**].
 - m** Select **Variables**.
 - n** Press **FCTN**.
 - o** Select **SAVE**. The frame positions and system variables are saved in the **SYSVAR.SV** file, on the default device.

Procedure 3-25 Selecting a User Frame

NOTE To select the number of the user frame you want to use, you can also use the jog menu. Refer to Section 2.2.8.

Condition ■ The user frame you want to select has been set up.

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select Frames
 - 5 If user frames are not displayed press F3, [OTHER], and select User Frame. If F3, [OTHER], is not displayed, press PREV. You will see a screen similar to the following.

SETUP Frames				JOINT	50%
User/RTCP	Setup/Direct	Entry	1/6		
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	*****	
Active UFRAME/RTCP \$MNUFRAMNUM[1]=0					
[TYPE] DETAIL [OTHER] CLEAR SETIND >					
[TYPE] CLRIND					

- 6 To select the user frame to use, press F5, SETIND, type the desired frame number and press ENTER. This sets the active user frame (\$MNUFRAMNUM[1]) to the number of the frame you want.

NOTE To select the number of the user frame you want to use, you can also use the jog menu. Refer to Section 2.2.8.

- 7** The system variable \$USEUFRAME defines whether the current value of \$MNUFRAMENUM[group_no] will be assigned to the position's user frame when it is being recorded or touched up.

When \$USEUFRAME=FALSE, the initial recording of positions and the touching up of positions is done with the user frame number equal to 0, regardless of the value of \$MNUFRAMENUM[group_no].

When \$USEUFRAME=TRUE, the initial recording of positions is done with the position's user frame equal to the user frame defined by \$MNUFRAMENUM[group_no]. The touching up of positions must also be done with the position's user frame equal to the user frame defined by \$MNUFRAMENUM[group_no].

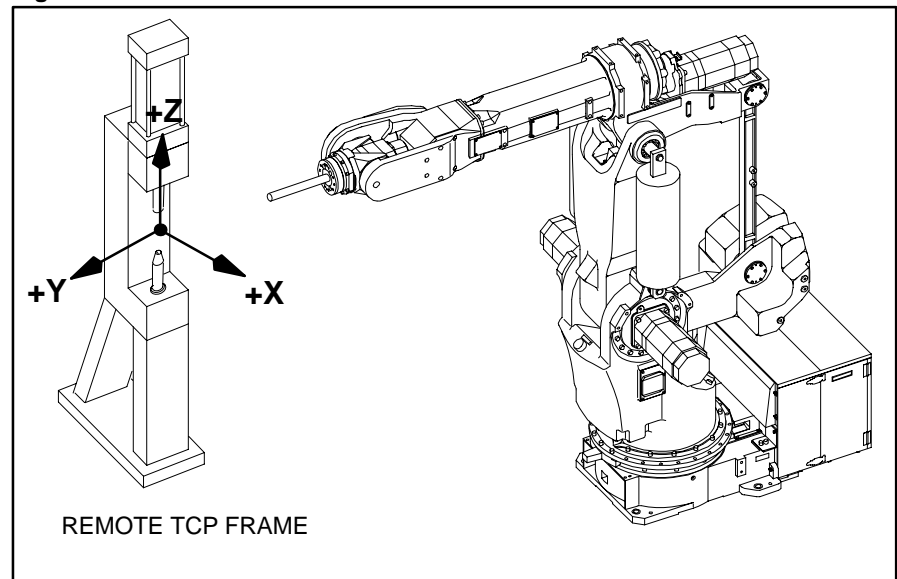
NOTE When a teach pendant program is executed, you must make sure that the user frame of the position equals the value of \$MNUFRAMENUM[group_no], otherwise, an error will occur. Set the value of \$MNUFRAMENUM[1] using the UFRAME_NUM=n instruction in the teach pendant program before you record the position to guarantee that the user frame numbers match during program execution.

3.8.3 Setting Up a Remote TCP Frame

You must define a remote TCP (RTCP) frame before you can use the remote TCP option to jog or include remote TCP within a motion instruction. Use Procedure 3-27 to set up an RTCP frame. You define this frame using the location of the remote TCP as the origin of the frame.

See Figure 3-32 for an example of a remote TCP frame.

Figure 3-32. Remote TCP Frame



You can use two methods to define the RTCP frame:

- Three point method
- Direct entry method

Three Point Method

Recording three points defines the RTCP frame. The three points are the origin, or TCP of the fixed tool, a position along the +x-axis of the RTCP frame, and a position on the y-axis of the RTCP frame. Use Procedure 3-26 to define an RTCP frame using the three point method.

Direct Entry Method

If you cannot use the three point method, use the direct entry method. In the direct entry method, you will be required to specify values for x, y, and z of the remote TCP. Use Procedure 3-27 to define an RTCP frame using the direct entry method.

Use Procedure 3-28 to select an RTCP frame.

Procedure 3-26 Setting Up a Remote TCP Frame Using the Three Point Method

**WARNING**

If you are setting up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.

If you are modifying an existing frame, make sure that all frame data is set the way you want before you change it.

Otherwise, you could injure personnel or damage equipment.

- Condition**
- The end-of-arm tooling or a setup pointer is attached to the robot faceplate.
 - You have set up the tool frame for the end-of-arm tooling or setup pointer. Refer to Section 3.8.1.
- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1 [TYPE].
 - 4 Select Frames.
 - 5 **If user/RTCP frames are not displayed**, press F3, [OTHER], and select User/RTCP. If F3, [OTHER], is not displayed, press PREV.
 - 6 **To display the settings for all frames**, press PREV repeatedly until you see a screen similar to the following.

SETUP Frames				JOINT 50%
User/RTCP Setup / Three Point				1/6
	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	*****
Active UFRAME/RTCP \$MNUFRAMNUM[1]=0				
[TYPE] DETAIL [OTHER] CLEAR SETIND >				
[TYPE] CLRIND >				

- 7 **To set the numerical values to zero**, move the cursor to the frame number and press F4, CLEAR.
- 8 Press F2, DETAIL.

Input frame number to display:

- 9 To select a frame,**
 - a** Press F3, FRAME.
 - b** Type the desired frame number.
 - c** Press ENTER.
- 10** Press F2, [METHOD].
- 11** Select Three Point. You will see a screen similar to the following.

```

SETUP Frames                                JOINT 50%
User/RTCP Setup/ Three Point                 1/4
Frame Number: 2
  X: 0.0    Y: 0.0    Z: 0.0
  W: 0.0    P: 0.0    R: 0.0

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

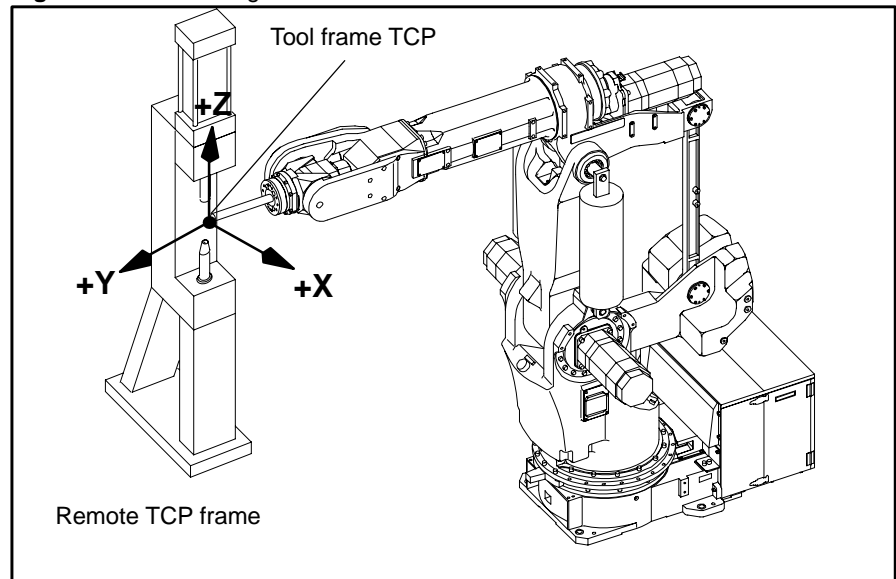
Active UFRAME/RTCP $MNUFRAMNUM[1]=0
[ TYPE ] [METHOD]  FRAME

```

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

13 Define the origin point of the remote TCP frame.**Orient Origin Point:** UNINIT

- a Move the cursor to Orient Origin Point.
- b Jog the robot so that the TCP of the end-of-arm tooling or setup pointer touches the remote TCP of the fixed tool. See Figure 3-33.
- c Press and hold the SHIFT key and press F5, RECORD.

Figure 3-33. Touching the TCP of the Robot Tool to the Remote TCP**14** Define the +X Direction Point:**X Direction Point:** UNINIT

- a Move the cursor to X Direction Point.
- b Select the WORLD coordinate system.
- c Jog the robot in the +x direction.
- d Press and hold the SHIFT key and press F5, RECORD.

15 Define a point on the positive X-Y plane:**Y Direction Point:** UNINIT

- a Move the cursor to Orient Origin Point.
- b Select the WORLD coordinate system.
- c Move the robot to the Orient Origin Point by pressing and holding the SHIFT key and pressing F4, MOVE_TO.
- d Move the cursor to Y Direction Point.
- e Jog the robot in the +y direction.
- f Press and hold the SHIFT key and press F5, RECORD.

16 To move to a recorded position, move the cursor to the desired position, press and hold the SHIFT key and press F4, MOVE_TO.

17 To select the RTCP frame to use, press F5, SETIND, type the number of the user frame you want, and press ENTER. This sets the active user frame (\$MNUFRAMNUM[1]) to the number of the frame you want.

NOTE To select the number of the RTCP frame you want to use, you can also use the jog menu. Refer to Section 2.2.8.

18 To clear the current RTCP frame selected, press NEXT, >, and then F2, CLRIND. This sets the active user frame (\$MNUFRAMNUM[1]) to zero, which means that the default user frame is currently selected.

 **CAUTION**

When you are finished setting the frame configuration, save the information to a storage device so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

- Saving Frame Configuration**
- 19 To save the frames and related system variables to a file** on the default device,
- a** Press MENUS.
 - b** Select FILE.
 - c** Press F1, [TYPE].
 - d** Select File.
 - e** Press F5, [UTIL].
 - f** Select Set Device.
 - g** Move the cursor to the device you want and press ENTER.
 - h** Press FCTN.
 - i** Select SAVE. This will save the frame positions and comments for all frames to the file, FRAMEVAR.SV, on the default device.
- Display the SYSTEM Variables menu,**
- j** Press MENUS.
 - k** Select SYSTEM.
 - l** Press F1, [TYPE].
 - m** Select Variables.
 - n** Press FCTN.
 - o** Select SAVE. The frame positions and system variables are saved in the SYSVAR.SV file, on the default device.

Procedure 3-27 Setting Up a Remote TCP Frame Using the Direct Entry Method

**WARNING**

If you are setting up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.

If you are modifying an existing frame, make sure that all frame data is set the way you want before you change it.

Otherwise, you could injure personnel or damage equipment.

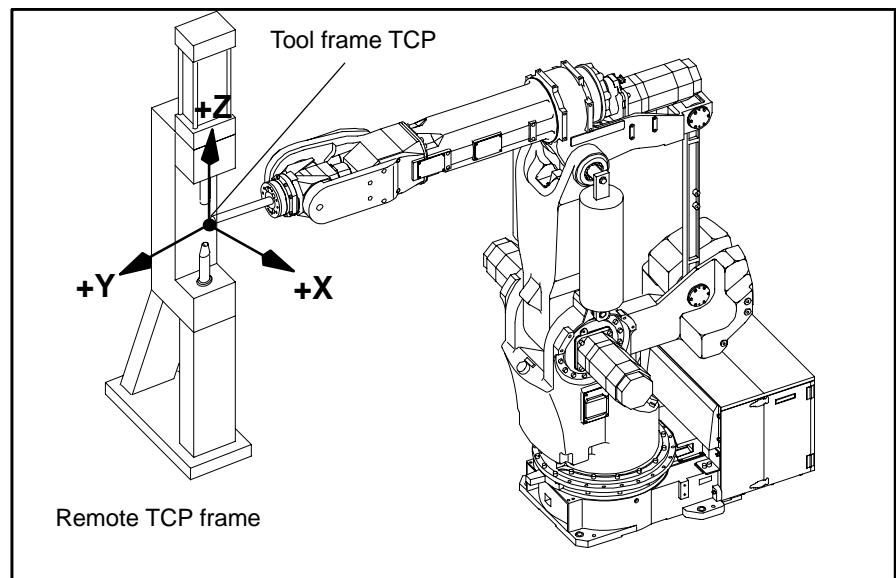
Condition

- The end-of-arm tooling or a setup pointer is attached to the robot faceplate.
- You have set up the tool frame for the end-of-arm tooling or setup pointer. Refer to Section 3.8.1.

Determine the Remote TCP of the Fixed Tool

- 1 Jog the robot so that the TCP of the end-of-arm tooling or setup pointer touches the remote TCP of the fixed tool. See Figure 3-34.

Figure 3-34. Touching the TCP of the Robot Tool to the Remote TCP



- 2 Press POSN.
- 3 Press F4, WORLD.
- 4 Write down the x, y, and z values of the current position in the WORLD coordinate system. This is the remote TCP of the fixed tool.

Use the Direct Entry Method to Set Up the Remote TCP Frame

- 5 Press MENUS.
- 6 Select SETUP.
- 7 Press F1, [TYPE].
- 8 Select Frames.
- 9 **If user/RTCP frames are not displayed**, press F3, [OTHER], and select User/RTCP. If F3, [OTHER], is not displayed, press PREV.
- 10 **To display the settings for all the frames**, press PREV repeatedly until you see a screen similar to the following.

SETUP Frames				JOINT	50%
User/RTCP	Setup/	Direct	Entry	1/6	
	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	*****	
Active UFRAME/RTCP \$MNUFRAMNUM[1]=0					
[TYPE] DETAIL [OTHER] CLEAR SETIND >					
[TYPE] CLRIND >					

- 11 **To set the numerical values to zero**, move the cursor to the frame number and press F4, CLEAR.
- 12 Press F2, DETAIL.
- 13 **To select a frame**,
 - a Press F3, FRAME.
 - b Type the desired frame number.
 - c Press ENTER.
- 14 Press F2, [METHOD].

Input frame number to display:

15 Select Direct Entry. You will see a screen similar to the following.

```

SETUP Frames                                JOINT 50%
User/RTCP Setup/ Direct Entry                1/7
Frame Number: 1
1 Comment:  *****
2 X:                0.000
3 Y:                0.000
4 Z:                0.000
5 W:                0.000
6 P:                0.000
7 R:                0.000
Configuration: N R D B, 0, 0, 0
Active UFRAME/RTCP $MNUFRAMENUM[1]=0
[ TYPE ] [METHOD] FRAME

```

16 To add a comment:

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

17 Set each position component:

- a Move the cursor to the x, y, and z components and enter the values you recorded in Step 4.
- b Move the cursor to the w, p, and r components and enter 0 for each component.

18 **To select the RTCP frame to use**, press F5, SETIND, type the number of the RTCP frame you want, and press ENTER. This sets the active user frame (\$MNUFRAMNUM[1]) to the number of the frame you want.

NOTE To select the number of the RTCP frame you want to use, you can also use the jog menu. Refer to Section 2.2.8.

19 **To clear the current frame to zero**, move the cursor to the frame number and press NEXT, >, and then F2, CLRIND. This sets the active user frame (\$MNUFRAMNUM[1]) to zero, which means that the default user frame is currently selected.

```

2 X                0.000
3 Y:                0.000
4 Z:                0.000
5 W:                0.000
6 P:                0.000
7 R:                0.000
Configuration: N R D B,0,0,0

```

⚠ CAUTION

When you are finished setting the frame configuration, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Save the Frame Configuration

20 To save the frames and related system variables to a file on the default device,

- a** Press **MENUS**.
- b** Select **FILE**.
- c** Press **F1**, [**TYPE**].
- d** Select **File**.
- e** Press **F5**, [**UTIL**].
- f** Select **Set Device**.
- g** Move the cursor to the device you want and press **ENTER**.
- h** Press **FCTN**.
- i** Select **SAVE**. This will save the frame positions and comments for all frames to the file, **FRAMEVAR.SV**, on the default device.

Display the SYSTEM Variables menu,

- j** Press **MENUS**.
- k** Select **SYSTEM**.
- l** Press **F1**, [**TYPE**].
- m** Select **Variables**.
- n** Press **FCTN**.
- o** Select **SAVE**. The frame positions and system variables are saved in the **SYSVAR.SV** file, on the default device.

Procedure 3-28 Selecting an RTCP Frame

Condition Step

NOTE To select the number of the RTCP frame you want to use, you can also use the jog menu. Refer to Section 2.2.8.

- The user frame you want to select has been set up.

- 1 Press **MENUS**.
- 2 Select **SETUP**.
- 3 Press **F1**, [**TYPE**].
- 4 Select **Frames**
- 5 If user frames are not displayed press **F3**, [**OTHER**], and select **User Frame**. If **F3**, [**OTHER**], is not displayed, press **PREV**. You will see a screen similar to the following.

SETUP Frames				JOINT	50%
User/RTCP	Setup/Direct	Entry	1 / 6		
	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	*****	
Active UFRAME/RTCP \$MNUFRAMNUM[1]=0					
[TYPE] DETAIL [OTHER] CLEAR SETIND >					
[TYPE] CLRIND					

- 6 To select the user frame to use, press **F5**, **SETIND**, type the number of the user frame you want, and press **ENTER**. This sets the active user frame ($\$MNUFRAMNUM[1]$) to the number of the frame you want.
- 7 The system variable $\$USEUFRAME$ defines whether the current value of $\$MNUFRAMENUM[group_no]$ will be assigned to the position's user frame when it is being recorded or touched up.
 - When $\$UFRAMENUM=FALSE$, the initial recording of positions and the touching up of positions is done with the user frame number equal to 0, regardless of the value of $\$MNUFRAMENUM[group_no]$.
 - When $\$UFRAMENUM=TRUE$, the initial recording of positions is done with the position's user frame equal to the user frame defined by $\$MNUFRAMENUM[group_no]$. The touching up of positions must also be done with the position's user frame equal to the user frame defined by $\$MNUFRAMENUM[group_no]$.

NOTE When a teach pendant program is executed, you must make sure that the user frame of the position equals the value of $\$MNUFRAMENUM[group_no]$, otherwise, an error will occur. Set the value of $\$MNUFRAMENUM[1]$ using the **UFRAME_NUM=n** instruction in the teach pendant program before you record the position to guarantee that the user frame numbers match during program execution.

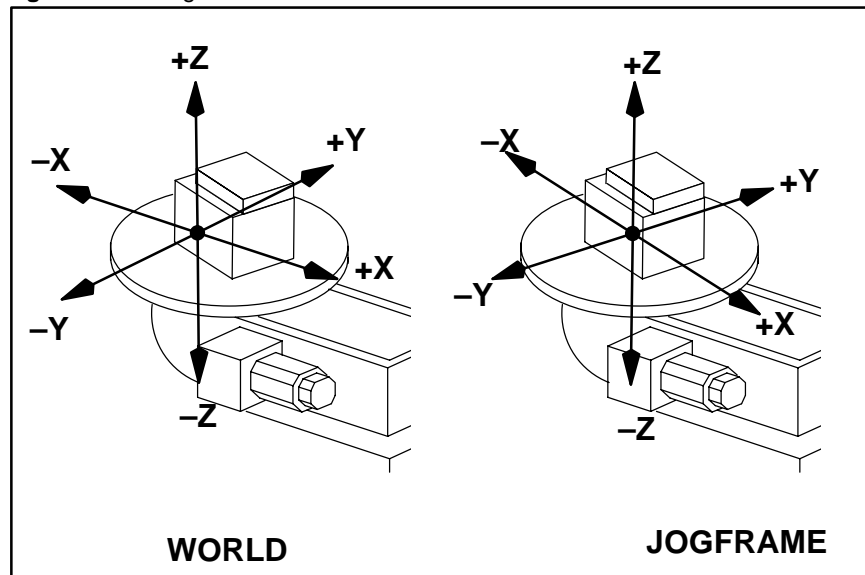
3.8.4 Setting Up Jog Frame

Jog frame is a frame that you can set up in any location, with any orientation. Jog frame provides a convenient way to move along a part when the part is oriented differently from the world frame. See Figure 3-35.

You can set up jog frame so that the coordinates of jog frame correspond to the coordinates of the part. You can then jog along x, y, and z to teach the positions on the part.

- Before you use jog frame, you must set up its location and orientation.
- You can set up as many as five different jog frames for each robot.
- You can select one jog frame to be active at a time.
- You can jog the robot in jog frame.

Figure 3-35. Jog Frame Defined Parallel to Part



You can use two methods to define the jog frame.

- Three point method
- Direct entry method

Three Point Method

The three point method allows you to define a jog frame by recording three points: the origin, a point along the +x-axis of the user frame, and a point on the x-y plane of the user frame (defines the x-y plane and the y-z plane). Use Procedure 3-29 to set up the jog frame using the three point method.

Direct Entry Method

The direct entry method allows you to designate the origin with values for x, y, z, w, p, and r. This method provides direct recording and numerical entry of the frame position. Use Procedure 3-30 to set up the jog frame using the direct entry method.

Use Procedure 3-31 to select a jog frame.

Procedure 3-29 Setting Up the Jog Frame Using the Three Point Method

**WARNING**

If you are setting up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.

If you are modifying an existing frame, make sure that all frame data is set the way you want before you change it.

Otherwise, you could injure personnel or damage equipment.

- Condition** ■ You have a cardboard box.
- Step**
- 1 Press MENU.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select Frames.
 - 5 **To choose the motion group** for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want: Group 1, Group 2, or Group 3. The default motion group is group 1.
 - 6 **If jog frames are not displayed**, press F3, [OTHER], and select Jog Frame. If F3, [OTHER], is not displayed, press PREV.
 - 7 **To display the settings for all frames**, press PREV repeatedly until you see a screen similar to the following.

SETUP Frames				JOINT 50%
JOG Frame Setup / Three Point				1/5
	X	Y	Z	Comment
1:	0.0	0.0	0.0	*****
2:	0.0	0.0	0.0	*****
3:	0.0	0.0	0.0	*****
4:	0.0	0.0	0.0	*****
5:	0.0	0.0	0.0	*****
Active JOG FRAME[1] = 0				
[TYPE] DETAIL [OTHER] CLEAR SETIND				

- 8 **To set the numerical values to zero**, move the cursor to the frame number and press F4, CLEAR.
- 9 Press F2, DETAIL.

Input frame number to display:

- 10 To select a frame,
 - a Press F3, FRAME.
 - b Type the desired frame number.
 - c Press ENTER.
- 11 Press F2, [METHOD].
- 12 Select Three Point. You will see a screen similar to the following.

```

SETUP Frames                                JOINT 50%
Jog Frame Setup / Three Point                 1/4
Frame number: 2

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

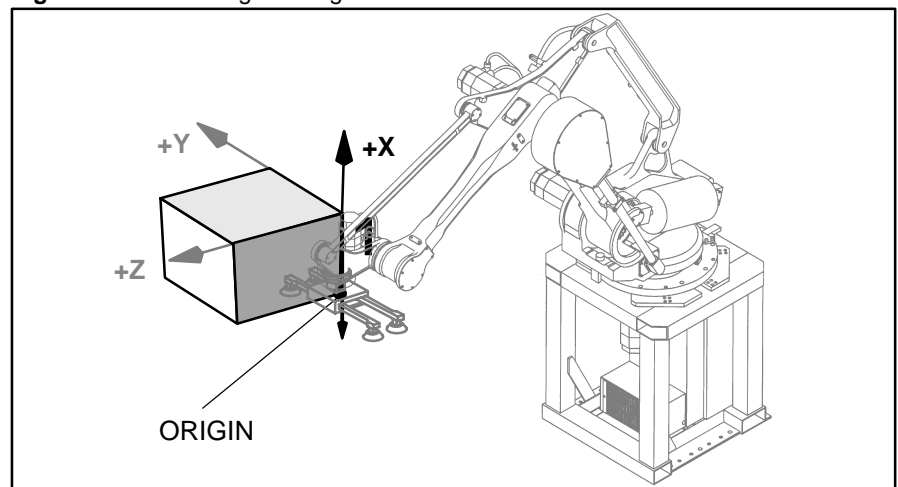
Comment: *****
Orient Origin Point: UNINIT
X Direction Point: UNINIT
Y Direction Point: UNINIT
Active JOG FRAME[i] = 0
[ TYPE ] [METHOD] FRAME

```

- 13 To add a comment:
 - a Move the cursor to the comment line and press ENTER.
 - b Select a method of naming the comment.
 - c Press the appropriate function keys to enter the comment.
 - d When you are finished, press ENTER.
- 14 Mount a box within the workcell so that the orientation of the box matches the orientation of the desired jog frame. Make sure that the corner of the box used to record the origin is at the proper location.
- 15 Define the origin of the jog frame.
 - a Move the cursor to System Origin Point.
 - b Jog the robot TCP to the origin. In Figure 3-36 the origin is labeled 1.
 - c Press and hold the SHIFT key and press F5, RECORD.

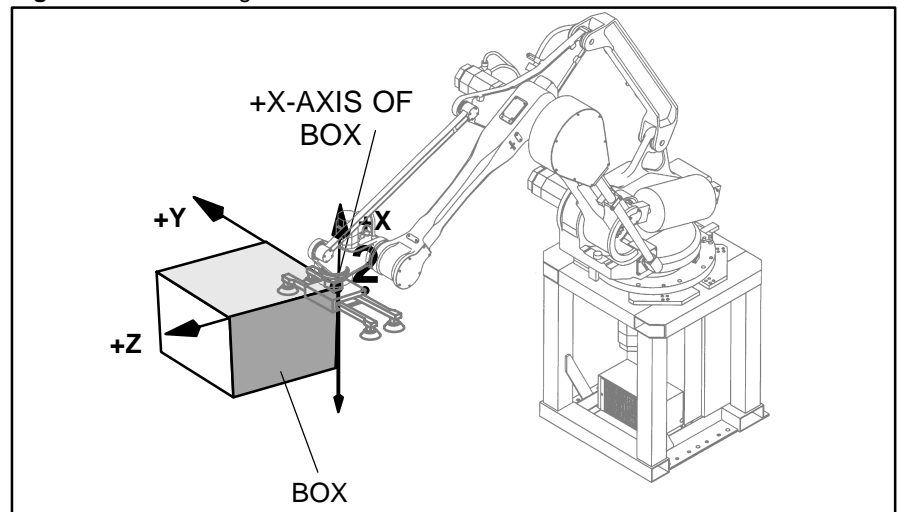
Orient Origin Point: **UNINIT**

Figure 3-36. Defining the Origin

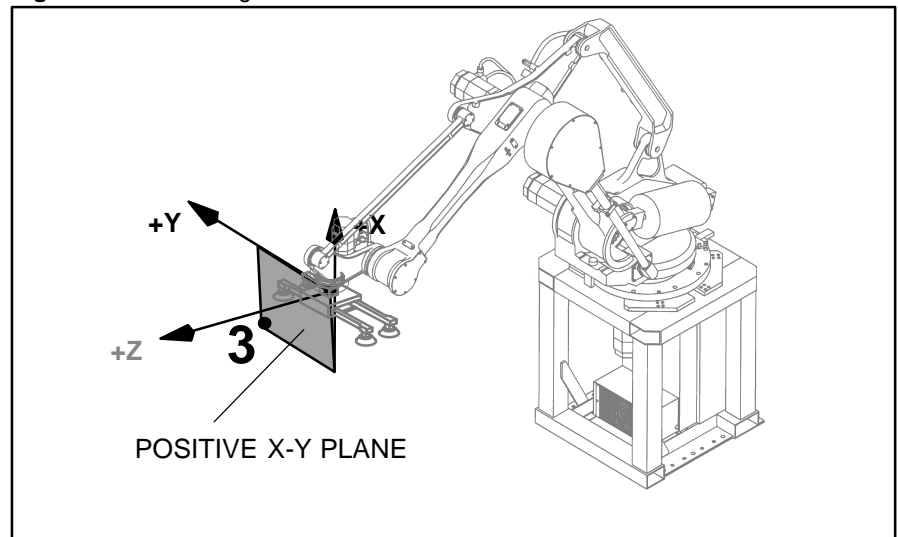


16 Define the +X Direction Point:**X Direction Point:** UNINIT

- a Move the cursor to X Direction Point.
- b Jog the robot along the x-axis of the box. In Figure 3-37 the origin is labeled 2.
- c Press and hold the SHIFT key and press F5, RECORD.

Figure 3-37. Defining the X Direction Point**17** Define a point on the positive X-Y plane:**Y Direction Point:** UNINIT

- a Move the cursor to Y Direction Point.
- b Jog the robot to a location on the positive X-Y plane. In Figure 3-38 this point is labeled number 3.
- c Press and hold the SHIFT key and press F5, RECORD.

Figure 3-38. Defining the X-Y Plane

- 18** To move to a recorded position, press and hold the SHIFT key and press F4, MOVE_TO.
- 19** To set the numerical values to zero, move the cursor to the frame number and press F4, CLEAR.
- 20** To select the jog frame to use, press F5, JGFRM, type the desired frame number and press ENTER.

NOTE To select the number of the jog frame you want to use, you can also use the jog menu. Refer to Section 2.2.8.

 **CAUTION**

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving Frame Configuration

- 21** To save the frames and related system variables to a file on the default device,
 - a** Press MENUS.
 - b** Select FILE.
 - c** Press F1, [TYPE].
 - d** Select File.
 - e** Press F5, [UTIL].
 - f** Select Set Device.
 - g** Move the cursor to the device you want and press ENTER.
 - h** Press FCTN.
 - i** Select SAVE. This will save the frame positions and comments for all frames to the file, FRAMEVAR.SV, on the default device.

Display the SYSTEM Variables menu,

 - j** Press MENUS.
 - k** Select SYSTEM.
 - l** Press F1, [TYPE].
 - m** Select Variables.
 - n** Press FCTN.
 - o** Select SAVE. The frame positions and system variables are saved in the SYSVAR.SV file, on the default device.

Procedure 3-30 Setting Up the Jog Frame Using the Direct Entry Method

**WARNING**

If you are setting up a new frame, make sure that all frame data is zero or uninitialized before you record any positions. Press F4, CLEAR, to clear frame data.

If you are modifying an existing frame, make sure that all frame data is set the way you want before you change it.

Otherwise, you could injure personnel or damage equipment.

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1 [TYPE].
 - 4 **To choose the motion group** for the frame you are setting up in systems with multiple motion groups press F3, [OTHER], and select the group you want: Group 1, Group 2, or Group 3. The default motion group is group 1.
 - 5 Select Frames.
 - 6 **If jog frames are not displayed**, press F3, [OTHER] and select Jog Frame. If F3, [OTHER], is not displayed, press PREV.
 - 7 **To display the settings for all frames**, press PREV repeatedly until you see a screen similar to the following.

SETUP Frames				JOINT	50%
JOG Frame Setup / Three Point				1/5	
	X	Y	Z	Comment	
1:	0.0	0.0	0.0	*****	
2:	0.0	0.0	0.0	*****	
3:	0.0	0.0	0.0	*****	
4:	0.0	0.0	0.0	*****	
5:	0.0	0.0	0.0	*****	
Active JOG FRAME[1] = 0					
[TYPE]		DETAIL	[OTHER]	CLEAR	SETIND

- 8 **To set the numerical values to zero**, move the cursor to the frame number and press F4, CLEAR.
- 9 Press F2, DETAIL.

Input frame number to display:

- 10 To select a frame,
 - a Press F3, FRAME.
 - b Type the desired frame number.
 - c Press ENTER.
- 11 Press F2, [METHOD].
- 12 Select Direct Entry. You will see a screen similar to the following.

```

SETUP Frames                                JOINT 50%
Jog Frame Setup / Direct Entry                1/7
Frame number: 1
 1 Comment *****
 2 X:                                0.000
 3 Y:                                0.000
 4 Z:                                0.000
 5 W:                                0.000
 6 P:                                0.000
 7 R:                                0.000
   Configuration:      N R D B, 0, 0, 0
Active JOG FRAME[i]=0
[ TYPE ] [METHOD]  FRAME  MOVE TO RECORD

```

- 13 To add a comment:
 - a Move the cursor to the comment line and press ENTER.
 - b Select a method of naming the comment.
 - c Press the appropriate function keys to enter the comment.
 - d When you are finished, press ENTER.
- 14 Set each position component:
 - a Move the cursor to the component.
 - b Enter the numeric value for the component.
 - c Press ENTER to set the new value.

```

2 X 0.000
3 Y: 0.000
4 Z: 0.000
5 W: 0.000
6 P: 0.000
7 R: 0.000
Configuration: N R D B,0,0,0

```

- 15 To select the jog frame to use, press F5, JGFRM, type the desired frame number and press ENTER.

NOTE To select the number of the jog frame you want to use, you can also use the jog menu. Refer to Section 2.2.8.

⚠ CAUTION

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

- Saving Frame Configuration**
- 16 To save the frames and related system variables to a file on the default device,**
- a** Press **MENUS**.
 - b** Select **FILE**.
 - c** Press **F1**, [**TYPE**].
 - d** Select **File**.
 - e** Press **F5**, [**UTIL**].
 - f** Select **Set Device**.
 - g** Move the cursor to the device you want and press **ENTER**.
 - h** Press **FCTN**.
 - i** Select **SAVE**. This will save the frame positions and comments for all frames to the file, **FRAMEVAR.SV**, on the default device.
- Display the SYSTEM Variables menu,**
- j** Press **MENUS**.
 - k** Select **SYSTEM**.
 - l** Press **F1**, [**TYPE**].
 - m** Select **Variables**.
 - n** Press **FCTN**.
 - o** Select **SAVE**. The frame positions and system variables are saved in the **SYSVAR.SV** file, on the default device.

Procedure 3-31 Selecting a Jog Frame

NOTE To select the number of the jog frame you want to use, you can also use the jog menu. Refer to Section 2.2.8.

Condition ■ The jog frame you want to select has been set up.

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select Frames.
 - 5 If jog frames are not displayed press F3, [OTHER], and select Jog Frame. If F3, [OTHER], is not displayed, press PREV. You will see a screen similar to the following.

SETUP Frames				JOINT 50%
Jog Frame Setup / Direct Entry				1/5
	X	Y	Z	Comment
1:	0.0	0.0	0.0	*****
2:	0.0	0.0	0.0	*****
3:	0.0	0.0	0.0	*****
4:	0.0	0.0	0.0	*****
5:	0.0	0.0	0.0	*****
Active JOG FRAME[1] = 0				
[TYPE]	DETAIL	[OTHER]	CLEAR	SETIND

- 6 To select the jog frame to use, press F5, SETIND, type the number of the jog frame you want, and press ENTER. This copies the selected jog frame to \$JOG_GROUP[group_no].\$JOGFRAME.

3.8.5 Saving Frame Data

Saving Frame data saves the frame positions and comments. Use Procedure 3-32 to save frame data to a file.

Procedure 3-32 Saving Frame Data to a File

- 1 Press MENUS.
- 2 Select SETUP.
- 3 Press F1, [TYPE].
- 4 Select Frames.
- 5 Press F2, DETAIL.
- 6 **To select a frame,**
 - a Press F3, FRAME.
 - b Type the desired frame number.
 - c Press, ENTER.
- 7 Press F2, [METHOD].
- 8 Select a frame method. You will see a screen similar to the following.

SETUP Frames				JOINT 50%
Tool	Frame Setup	/ Three Point		1/5
	X	Y	Z	Comment
1:	0.0	0.0	0.0	*****
2:	0.0	0.0	0.0	*****
3:	0.0	0.0	0.0	*****
4:	0.0	0.0	0.0	*****
5:	0.0	0.0	0.0	*****
ACTIVE TOOL \$MNUTOOLNUM[1]=1				
[TYPE] DETAIL [OTHER] CLEAR SETIND				

⚠ CAUTION

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

Saving Frame Configuration

- 9 To save the frames and related system variables to a file** on the default device,
 - a** Press **MENUS**.
 - b** Select **FILE**.
 - c** Press **F1**, [**TYPE**].
 - d** Select **File**.
 - e** Press **F5**, [**UTIL**].
 - f** Select **Set Device**.
 - g** Move the cursor to the device you want and press **ENTER**.
 - h** Press **FCTN**.
 - i** Select **SAVE**. This will save the frame positions and comments for all frames to the file, **FRAMEVAR.SV**, on the default device.

Display the SYSTEM Variables menu,

 - j** Press **MENUS**.
 - k** Select **SYSTEM**.
 - l** Press **F1**, [**TYPE**].
 - m** Select **Variables**.
 - n** Press **FCTN**.
 - o** Select **SAVE**. The frame positions and system variables are saved in the **SYSVAR.SV** file, on the default device.

3.9 PRODUCTION OPERATION SETUP

Production operation setup allows you to set up a program so that it is run automatically during production.

To run production you can use

- Robot Service Request (RSR)
- Program Number Select (PNS)
- UOP PRODUCTION START input
- SOP CYCLE START input

This section includes information on how to set up RSR and PNS programs. UOP PRODUCTION START and SOP CYCLE START inputs do not require any software setup. Refer to Chapter 7 for more information about running production.

3.9.1 Robot Service Request (RSR)

A Robot Service Request (RSR) is a request for service from an external device. That request comes from a digital input signal on a preassigned RSR input line. You can use up to four robot service request signals: RSR1, RSR2, RSR3, and RSR4.

When the robot controller receives a service request signal, the controller determines whether the signal is acceptable. If acceptable, the controller determines which program to execute.

CAUTION

Any program that you want to execute by using RSRs must be named RSR[nnnn], where [nnnn] represents a four digit number from 0001 to 9999; otherwise, the program will not be executed.

If no other program is currently running, the program assigned to the RSR input line starts. If a program is currently running, the robot stores the signal and runs the program when the other program is finished.

When the robot receives the RSR signal, the robot can output the corresponding acknowledge signals (ACK1 – ACK4) if the signals are enabled.

Table 3-13 lists and describes each RSR setup item.

Table 3-13. RSR Setup Item Description

RSR SETUP ITEM	DESCRIPTION
RSR or PNS	This item allows you to specify the kind of production operation you want: RSR or PNS.
RSR1 Program Number	This item allows you to enter a number that when added to the base number defines the program number that will be executed when the RSR1 signal is received. For example, if you entered 23 for the RSR1 program number and the base number was set to 100, the RSR1 signal would execute program RSR0123. If you enter an invalid program number, the system will ignore this signal.
RSR2 Program Number	This item allows you to enter a number that when added to the base number defines the program number that will be executed when the RSR2 signal is received. If you enter an invalid program number or a zero, the system will ignore this signal.
RSR3 Program Number	This item allows you to enter a number that when added to the base number defines the program number that will be executed when the RSR3 signal is received. If you enter an invalid program number or a zero, the system will ignore this signal.
RSR4 Program Number	This item allows you to enter a number that when added to the base number defines the program number that will be executed when the RSR4 signal is received. If you enter an invalid program number or a zero, the system will ignore this signal.
Base Number	This item allows you to enter a number that when added to the RSR1-4 program number defines which program will be executed. This base number can be changed from within your program by using the PARAMETER NAME instruction. The parameter that contains the RSR base number is \$SHELL_CFG.\$JOB_BASE . By changing the base number, you can control which group of programs will be executed.
Acknowledge Function	This item allows you to enable or disable robot acknowledge output signals ACK1-4. FALSE means the signals are disabled. TRUE means the signals are enabled.
Acknowledge Pulse Width	This item allows you to set the length of the ACK1-4 signal when the acknowledge function is enabled.

Use Procedure 3-33 to set up RSRs.

Procedure 3-33 RSR Setup

- Condition**
- UOP signals must be installed and configured.
 - The program name must be RSR[nnnn] where [nnnn] represents a four digit number from 0001 to 9999.

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select RSR/PNS. You will see a screen similar to the following.

RSR/PNS		JOINT	50 %
			1/8
1	RSR or PNS	[RSR]	
2	RSR1 program number	[ENABLE]	[12]
3	RSR2 program number	[ENABLE]	[3]
4	RSR3 program number	[ENABLE]	[18]
5	RSR4 program number	[ENABLE]	[64]
6	Base number		[100]
7	Acknowledge function		[FALSE]
8	Acknowledge pulse width(msec)		[10]
Power OFF then ON to enable changes.			
	[TYPE]	PNS	RSR

- 5 If RSR is not already displayed on line 1, press F5, RSR.
- 6 Move the cursor to the item you want to set and enter the value.

**WARNING**

When you change the kind of production operation, you must turn off the controller and turn the controller on to use the new information; otherwise, the new settings can not be accepted.

3.9.2 Program Number Select (PNS)

A Program Number Select (PNS) is a method of selecting the name of a program to be run by some external device. The name of the program to be run comes as a group of input signals from an external device on eight PNS input lines.

The following sequence takes place with PNS operation:

CAUTION

Any program that you want to execute by using PNS must be named PNS[nnnn], where [nnnn] represents a four digit number from 0001 to 9999; otherwise, the program will not be executed.

1. The eight PNS inputs signal a binary number to the system.
2. The binary number is added to the base number if a base number is used. Refer to Table 3-14. This defines the program number to be executed and makes that program the default program.
3. SNO 1-8 is loaded with the binary number of the original eight PNS inputs.
4. SNACK is pulsed to signal the external device to read, ACK1-8.
5. The PLC can use ACK1-8 and SNACK to check the PNS number. If the number received on SNO 1-8 is the same as the number sent out on PNS 1-8, the PROD_START input signal is sent to the controller.
6. The robot will run the program when the PROD_START input signal is received.

PNS signals can be used for multi-tasking. Once a program has started running, PNS signals and the START input can be used to execute a second program. The system variable \$SHELL_CFG.\$CONT_ONLY must be set to FALSE to allow the START input to execute the currently selected program.

Table 3-14 lists and describes each PNS setup item.

Table 3-14. PNS Setup Item Description

PNS SETUP ITEM	DESCRIPTION
RSR or PNS	This item allows you to specify the kind of production operation you want: RSR or PNS.
Base Number	This item allows you to enter a number that when added to the PNS1-8 binary signal defines which program will be executed. For example, if the PNS1-8 input is 0023, and the base number is 100, then PNS0123 will be executed. This base number can be changed from within your program by using the PARAMETER NAME instruction. The parameter that contains the base number is \$\$HELL_CFG.\$PNS_BASE . By changing the base number, you can control which program will be executed.
Acknowledge Pulse Width	This item allows you to set the length of the SNACK1-8 signals.

Use Procedure 3-34 to set up PNS.

Procedure 3-34 PNS Setup

Condition ■ UOP signals must be installed and configured.

- Step**
- 1 Press **MENUS**.
 - 2 Select **SETUP**.
 - 3 Press **F1**, [TYPE].
 - 4 Select **RSR/PNS**. You will see a screen similar to the following.

RSR/PNS	JOINT	50 %
		1/3
1 RSR or PNS	[PNS]
2 Base number	[100]
3 Acknowledge pulse width(msec)	[10]
Power OFF then ON to enable changes.		
[TYPE]	PNS	RSR

- 5 If **PNS** is not already displayed on line 1, press **F4**, **PNS**.
- 6 Move the cursor to the item you want to set and enter the value.



WARNING

When you change the kind of production operation, you must turn off the controller and turn the controller on to use the new information; otherwise, the new settings can not be accepted.

3.10 MACRO COMMANDS

A macro command program is a separate program that contains a series of instructions to perform a task, and specify to run when

- A teach pendant key is pressed
- An item on the MANUAL FCTNS menu is selected
- An instruction in a program is executed

To use a macro command, you must

- Write the macro command program
- Set up the macro command to define how it will be executed
- Execute the macro command program

This section describes how to set up macro commands from the teach pendant, MANUAL FCTNS Macros screen. Refer to Section 6.17 for information on using macro command instructions in a program. Refer to Section 3.10.2 for information on executing the macro command.

3.10.1 Setting Up Macro Commands

Macro commands must be set up before they can be used. You can set them to run from a teach pendant user key, from the MANUAL FCTNS screen.

Teach Pendant User Keys

You can set up a macro command to run when a teach pendant user key is pressed alone or with the SHIFT key. If you want to execute a program that contains robot motion when a user key is pressed, you must set it up to run when the SHIFT key is pressed.

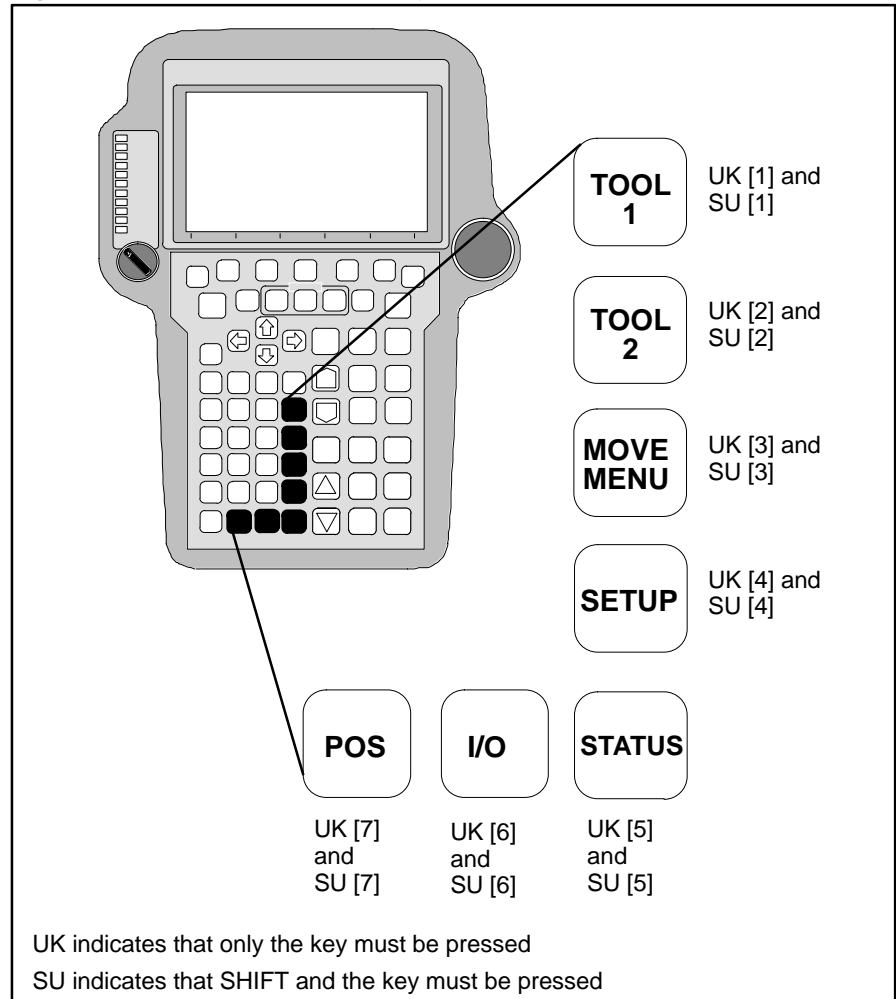
 **CAUTION**

Make certain that your application has not already assigned functions to the teach pendant user keys; otherwise, execution problems can occur.

When you set up macro commands, you can define up to seven macro commands to run when the user key is pressed alone (UK[1] – UK[7]), and seven macro commands to run when the user key is pressed with the SHIFT key (SU[1] – SU[7]). The macro commands that require the user key to be pressed alone (UK[1] – UK[7]) cannot contain any instructions that move the robot, and the group mask must be set to [*,*,*,*] in the program header information.

See Figure 3-39 for the location of these keys.

Figure 3-39. Teach Pendant User Keys



MANUAL FCTNS Macro Screen Items

You can set up a macro command program to be executed from the MANUAL FCTNS Macros screen. When you set up a macro command to run from this screen, selecting a manual functions menu item and pressing SHIFT and the EXEC function key executes the macro command. Refer to Procedure 3-35 to execute a macro command from the MANUAL FCTNS menu.

Input Signals

You can set up a macro command program to be executed when the input signal you specify is received. You can assign a macro command to a digital input (DI), robot input (RI), or User Operator Panel input (UI).

For **digital input signals**, indexes 0 through 99 are available. An index of 0 indicates that no macro is assigned. You can assign any of these index numbers to the macro command, but the digital signal must be configured properly for the macro command to execute.

For **robot input signals**, indexes 0 through the number of robot input signals configured on your system are available. An index of 0 indicates that no macro is assigned.

For **UOP input signals**, indexes 0 through the number of UOP input signals configured on your system are available. An index of 0 indicates that no macro is assigned.

Use Procedure 3-35 to set up a macro command.



WARNING

Before copying a program with embedded macros from one controller to another, compare the Setup menu macro lists of the two controllers. Be sure that the list on the first controller matches the list on the second controller. If they are not identical, DO NOT copy the program; otherwise, unexpected results could occur.

Procedure 3-35 Setting Up a Macro Command

- Condition**
- A macro program has been created.
 - The macro program has been tested and runs properly.
- Step**
- 1 Press **MENUS**.
 - 2 Select **SETUP**.
 - 3 Press **F1**, **[TYPE]**.
 - 4 Select **Macro**. You will see a screen similar to the following.

Macro Command		JOINT	10 %
Instruction name	Program	Assign	
1 []	[]	[]	[0]
2 [Hand open][hndopen1]	UK	[1]
3 [Hand close][hndclsel]	MF	[4]
4 []	[]	[]	[0]
5 []	[]	[]	[0]
6 []	[]	[]	[0]
7 []	[]	[]	[0]
8 []	[]	[]	[0]
9 []	[]	[]	[0]
10 []	[]	[]	[0]

[TYPE] CLEAR

```
1 [ ] [ ] [ 0 ]
```

- 5 Move the cursor to a blank **Instruction name** and press **ENTER**.

- 6 Name the instruction.
 - a Select a method of naming.
 - b Press the appropriate function keys to enter a name.
 - c When you are finished press **ENTER**.

```
1 [Start process ] [ ] [ 0 ]
```

- 7 Move the cursor to **Program** and press **F4**, **[CHOICE]**.

- 8 Select the macro program you want assigned to the instruction name and press **ENTER**.

```
1 [Start program ][prog01 ] [ 0 ]
```

- 9 Move the cursor to **Assign** and press **F4**, **[CHOICE]**.

10 Select the macro command assignment you want and press ENTER:

- For a user key without **SHIFT**, select UK.
- For a user key with **SHIFT**, select SU.
- For a **MANUAL FCTNS menu item**, select MF.
- For a **digital input**, select DI.
- For a **robot input**, select RI.
- For a **UOP input**, select UI.
- To remove an assignment, select --.

```
1 [Start program ][prog01 ]UK[1]
```

11 Move the cursor to the assignment number, enter the number, and press ENTER.

12 If you want to modify an entry, move the cursor to the item you want to change and enter a new value (or, press the **SHIFT** key and **F2**, **CLEAR** to remove the current value from the item and then begin typing).

 **CAUTION**

When all I/O is configured, save the information to a default device (disk) so that you can reload the configuration data if necessary. Otherwise, if the configuration is altered, you will have no record of it.

13 To save the information

- a Press **FCTN**.
- b Select **SAVE**. The file will be saved to the **SYSMACRO.SV** file on the default device. Refer to Chapter 9 for information on setting the device.

3.10.2 Executing Macro Commands

After you have set up macro commands you can run them, using one of the following methods:

- Press a teach pendant user key
- Press a teach pendant user key with the SHIFT key
- Select a MANUAL FCTNS Macros screen item
- Execute a macro program from within another program using the macro command instruction
- Receive an input signal (DI, RI, or UI)
- Execute a macro program

The method you use depends on how you set up the macro command to execute. This section describes how to execute a macro command from a teach pendant user key, and the MANUAL FCTNS Macros screen.



WARNING

Before copying a program with embedded macros from one controller to another, compare the Setup menu macro lists of the two controllers. Be sure that the list on the first controller matches the list on the second controller. If they are not identical, DO NOT copy the program; otherwise, unexpected results could occur.

Teach Pendant User Keys

Use Procedure 3-36 to execute a macro command that has been assigned to a teach pendant user key.

Procedure 3-36 Executing a Macro Command from a Teach Pendant User Key

- | | |
|------------------|---|
| Condition | <ul style="list-style-type: none"> ■ The program you want to use as a macro command has been tested. ■ The macro command has been set up to execute when a teach pendant user key is pressed. |
| Step | <ol style="list-style-type: none"> 1 Make sure the teach pendant is ON and the DEADMAN switch is pressed. |



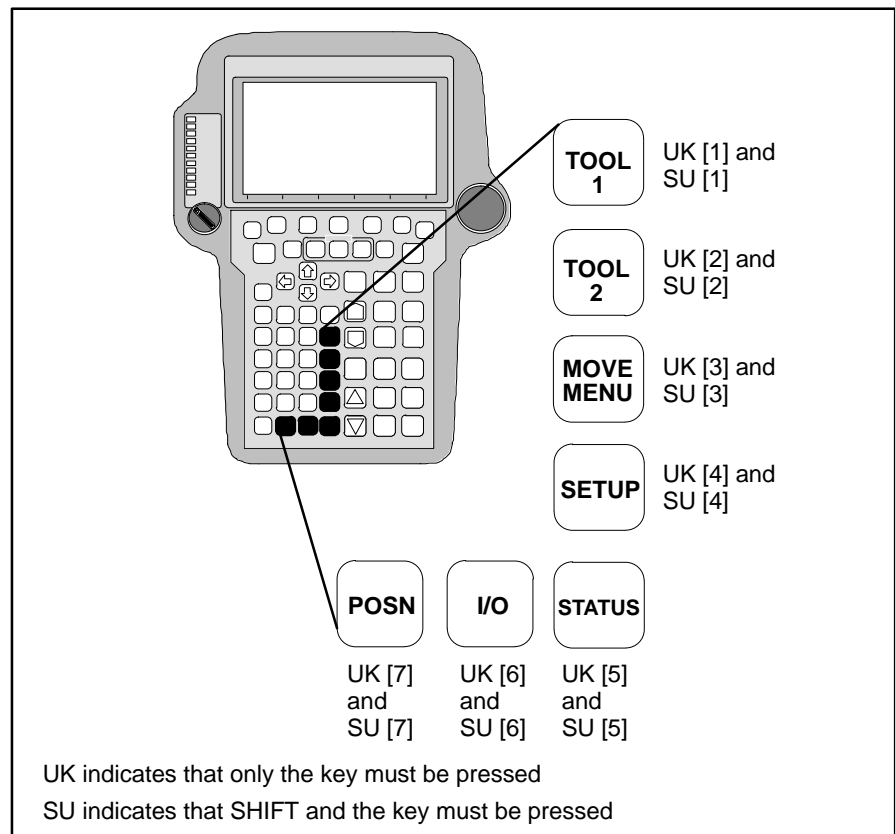
WARNING

In the next step, the robot could move. Make sure that personnel and unnecessary equipment are out of the workcell; otherwise, personnel could be injured and equipment damaged.

- 2 Press the teach pendant user key that corresponds to the macro command you assigned. If you assigned the key to be pressed with the SHIFT key, press and hold SHIFT and press the user key.

See Figure 3-40.

Figure 3-40. Teach Pendant User Keys



Teach Pendant Keys TOOL1 and TOOL2

The HandlingTool software provides six predefined macro commands:

- Open hand 1
- Close hand 1
- Relax hand 1
- Open hand 2
- Close hand 2
- Relax hand 2

All of the hand 1 macros have been defined to work with the teach pendant key TOOL 1. All of the hand 2 macros have been defined to work with the teach pendant key TOOL 2.

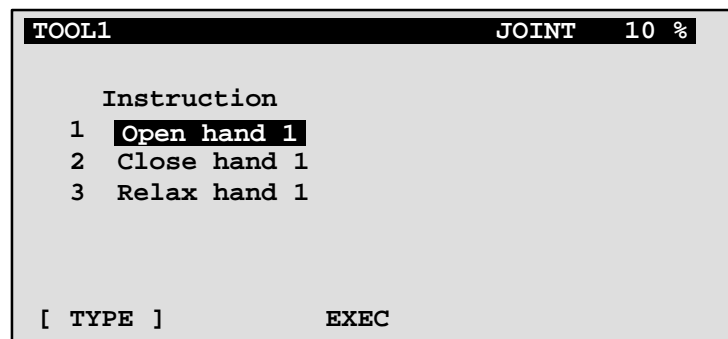
You can set up more macro commands for anything you would like to be able to do manually, for example, to repair or maintain equipment. Refer to Section 3.10 for more information about setting up macro commands.

Before you can use these six pre-defined macro commands, you must:

- Create and write a macro program for each macro command.
- Make sure the macro programs control the RI/RO so that the hand
 - Opens
 - Closes
 - Relaxes
- Assigned each macro program to its corresponding macro command. You do this by using the Macros screen. Refer to Section 3.10.1.

Procedure 3-37 Using the Teach Pendant Keys TOOL1 and TOOL2

- Condition**
- A macro program has been created and written for each pre-defined macro command that you want to use.
 - The macro programs have been assigned to their corresponding macro commands.
- Step**
- 1 Press either TOOL1 or TOOL2. You will see a screen similar to the following.



- 2 Move the cursor to the macro command you want to perform.
- 3 Press and hold in the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
- 4 Press and hold the SHIFT key and press F3, EXEC. The F3 key can then be released, but the SHIFT key must be held continuously until the instruction has completed executing.

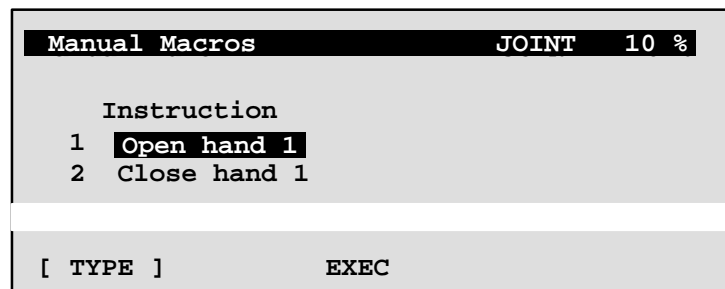
MANUAL FCTNS Menu Items

Use Procedure 3-38 to execute a macro command that has been assigned to a MANUAL FCTNS menu item. Refer to Section 3.10 for Macro setup.

Procedure 3-38 Executing a Macro Command from the MANUAL FCTNS Menu

- Condition**
- The program you want to use as a macro command has been tested.
 - The macro command has been set up to execute when an item on the MANUAL FCTNS screen is selected.

- Step**
- 1 Press MENU.
 - 2 Select MANUAL FCTNS. You will see a screen similar to the following.



- 3 Select an item on the menu.
- 4 Continuously press and hold in the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

**WARNING**

In the next step, the robot could move. Make sure that personnel and unnecessary equipment are out of the workcell; otherwise, personnel could be injured and equipment damaged.

- 5 Press and hold the SHIFT key and press F3, EXEC. The F3 key can be released, but the SHIFT key must be held continuously until the instruction has completed executing.

NOTE If the SHIFT key is released, the Macro program is aborted and cannot be resumed.

3.11 AXIS LIMITS SETUP

Axis limits define the motion range of the robot. The operating range of the robot axes can be restricted because of:

- Work area limitations
- Tooling and fixture interference points
- Cable and hose lengths

There are three methods used to prevent the robot from going beyond the necessary motion range. These are

- Axis limit software settings
- Axis limit switches
- Axis limit hardstops



WARNING

Do not use axis software limits as the only method for restricting robot motion. Modify the hard stops to match the software modifications; otherwise, you could injure personnel or damage equipment.

Software Settings

Axis limit **software settings** are upper and lower motion degree limitations. The limits can be set for all robot axes and will stop robot motion if the robot is calibrated. If the robot is not calibrated, overtravel limit switches are contacted two to three degrees beyond the software limits. Overtravel switches are used only on the major axes.

Limit Switches

Axis **limit switches** are overtravel switches that, when tripped, cut power to the servo motors. These are located two or three degrees beyond the software limits. Overtravel switches are used only on the major axes.

Hardstops

Axis limit **hardstops** are physical barriers that are located two or three degrees beyond the overtravel limit switch on the three major axes. The robot cannot move beyond a hardstop.

Setting the axis limits software settings changes the motion range of the robot. The axis limit screen displays the current upper and lower axis limits, for each robot axis, in degrees.

Upper Limits

Displays the upper limits of each axis, or the axis limits in a positive direction.

Lower Limits

Displays the lower limits of each axis, or the axis limits in a negative direction.

Saving Limits

After you change the axis limits, turn off the controller and then turn it on again so the new settings can be used.

⚠ CAUTION

Changing the axis limits will affect the robot work area, and could change robot motion. Anticipate the effects of changing axis limits before changing them; otherwise, unexpected results could occur, such as error in previously recorded positions.

Use Procedure 3-39 to set up axis limits.

Procedure 3-39 Setting Up Axis Limits

- Step**
- 1 Press **MENUS**.
 - 2 Select **SYSTEM**.
 - 3 Press **F1**, [**TYPE**].
 - 4 Select **Axis Limits**. You will see a screen similar to the following.

System Axis Limits				JOINT	100%
AXIS	GROUP	LOWER	UPPER		1/16
1	1	-150.00	150.00		dg
2	1	-60.00	100.00		dg
3	1	-110.00	50.00		dg
4	1	-240.00	240.00		dg
5	1	-120.00	120.00		dg
6	1	-450.00	450.00		dg
7	1	-20.00	1000.00		m
8	0	0.00	0.00		m
9	0	0.00	0.00		m
					0.00 m

[TYPE]

NOTE 0 indicates the robot does not have these axes.

- 5 Move the cursor to the axis limit you want to set.



WARNING

Do not depend on axis limit software settings to control the motion range of your robot. Use the axis limit switches and hardstops also; otherwise, injury to personnel or damage to equipment could occur.

- 6 Type the new value using the numeric keys on the teach pendant.
- 7 Repeat Steps 5 through 6 until you are finished setting the axis limits.



WARNING

You must turn off the controller and then turn it back on to use the new information; otherwise, injury to personnel or damage to equipment could occur.

- 8 Turn off the controller and then turn it back on again so the new information can be used.

3.12 BRAKE TIMERS SETUP

Brake timers define the length of time the robot remains idle before the brakes are applied. Brake timers are specified in milliseconds. For example, if you want the timer to be set to 2 seconds, you must set it to 2000.

After you set the brake timers, you must turn off the controller and then turn it back on again so the new information can be used.

Use Procedure 3-40 to set brake timers.

Procedure 3-40 Setting Brake Timers

- Step**
- 1 Press MENUS.
 - 2 Select SYSTEM.
 - 3 Press F1, [TYPE].
 - 4 Select Variables. You will see a screen similar to the following.

SYSTEM Variables		JOINT	50%
			1/168
1	\$ANGTOL	[9] of REAL	
2	\$APPLICATION	[3] of STRING [21]	
3	\$AP_MAXAX	0	
4	\$AP_PLUGGED	2	
5	\$AP_TOTALAX	16777216	
6	\$AP_USENUM	[32] of BYTE	
7	\$ASCII_SAVE	FALSE	
8	\$AUTOINIT	2	
9	\$AWECFG	AWECFG_T	
10	\$AWEOFT	AWEOTF_T	
	[TYPE]		

To move quickly through the information, press and hold down the SHIFT key and press the down or up arrow keys.

```
$SCR_GRP[1]  SCR_GRP_T
```

```
$BRK_NUMBER [9] of BYTE
```

1	[1]	1
2	[2]	2
3	[3]	2
4	[4]	3
5	[5]	3
6	[6]	3
7	[7]	0
8	[8]	0
9	[9]	0

```
$PARAM_GROUP MRR_GRP_T
```

```
$SV_OFF_ENB [9] of BOOLEAN
```

1	[1]	TRUE
2	[2]	TRUE
3	[3]	TRUE
4	[4]	TRUE
5	[5]	TRUE
6	[6]	TRUE
7	[7]	FALSE
8	[8]	FALSE
9	[9]	FALSE

5 Determine which brakes control each axis:

- Move the cursor to \$SCR_GRP and press ENTER.
- If you have more than one motion group, select the motion group number of the axes and press ENTER.
- Move the cursor to SCR_GRP_T and press ENTER.
- Move the cursor to \$BRK_NUMBER and press ENTER.
- Determine the brakes that control each axis.

The number in the left column is the axis number. The number in the far right column is the brake number. For example, the brake that controls axis 3 is brake number 1.

Some brakes control multiple axes. For example, axes 1, 2, 3, 4, 5, and 6 are all controlled by brake number 1. If the brake number is 0, no brakes control the axis.

- Press PREV **three times**, or until the first system variable screen is displayed.

6 Make sure that the brakes are enabled for the axes you want to control:

- Move the cursor to \$PARAM_GROUP and press ENTER.
- If you have more than one motion group, select the motion group number of the axes and press ENTER.
- Move the cursor to MRR_GRP_T and press ENTER.
- Move the cursor to \$SV_OFF_ENB and press ENTER.
- Move the cursor to the axis you want to check and set.

The number in the left column is the axis number. The number in the far right column is the value. For example, the value for axis 3 is TRUE.

If the value is TRUE, the brakes are enabled for the axis and you can define a brake timer.

If the value is FALSE, the brakes are not enabled for the axis and you cannot define a brake timer. Press F4, TRUE and press ENTER.

- Press PREV **three times**, or until the first system variable screen is displayed.

```
$PARAM_GROUP      MPR_GRP_T
```

```
$SV_OFF_TIME [9] of INTEGER
```

```
1 [1] 3000
2 [2] 3000
3 [3] 3000
4 [4] 3000
5 [5] 3000
6 [6] 3000
7 [7] 3000
8 [8] 3000
9 [9] 3000
```

7 Set the brake timer for the axes you want:

- a** Move the cursor to \$PARAM_GROUP and press ENTER.
- b** If you have more than one motion group, select the motion group number of the axes and press ENTER.
- c** Move the cursor to MPR_GRP_T and press ENTER.
- d** Move the cursor to \$SV_OFF_TIME and press ENTER.

The number in the left column is the axis number. The number in the right column is the time the robot remains idle before brakes are applied.

- e** Select an axis, type the new time (in milliseconds), and press ENTER.

NOTE If the same brake controls multiple axes, and you set brake timers for more than one of these axes, the **shortest** brake timer is effective.



WARNING

You must turn off the controller and then turn it back on to use the new information; otherwise, injury or damage to equipment could occur.

- 8** Turn off the controller. Then turn it back on so the new information can be used.

3.13 BRAKE ON HOLD SETUP

Brake on Hold defines whether the robot brakes are engaged (enabled) or disengaged (disabled) when the robot is placed in a hold condition. The available settings are summarized in Table 3-15. Use Procedure 3-41 to set brake on hold.

Table 3-15. Brake On Hold Settings

BRAKE ON HOLD SETTING	DESCRIPTION
Disabled	The brakes are not applied when the robot is in a hold condition.
Enabled	The brakes are applied when the robot is in a hold condition after a period of time.



WARNING

Not all axes have brakes. Enabling Brake on Hold has NO EFFECT on axes that do not have brakes. Make certain that you understand which axes have brakes before you enable Brake on Hold; otherwise, injury can result.

Procedure 3-41 Setting Brake On Hold

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select General. You will see a screen similar to the following.

SETUP General	JOINT	100 %
		1 / 4
1 Brake on hold:	DISABLED	
2 Current language:	DEFAULT	
3 Ignore Offset command:	DISABLED	
4 Ignore Tool_offset:	DISABLED	
[TYPE]	ENABLE	DISABLE

- 5 Move the cursor to Brake on hold.

NOTE Brake on Hold is disabled by default.

- 6 Enable or disable the brake on hold:
 - To enable the brake on hold, press F4, ENABLE.
 - To disable the brake on hold, press F5, DISABLE.

3.14 CURRENT LANGUAGE SETUP

Current language allows you to change the current language. You can select from only those languages that have dictionaries.

Use Procedure 3-42 to set the current language.

Procedure 3-42 Setting Current Language

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select General. You will see a screen similar to the following.

SETUP General		JOINT	100 %
			1 / 4
1 Brake on hold:	DISABLED		
2 Current language:	DEFAULT		
3 Ignore Offset command:	DISABLED		
4 Ignore Tool_offset:	DISABLED		
[TYPE]	ENABLED	DISABLED	

- 5 Select Current language.
- 6 Press F4, [CHOICE].
- 7 Select the language.

3.15 IGNORE OFFSET SETUP

This function ignores the position offset specified by an OFFSET instruction.

- When this function is disabled, the robot moves to the position to which a position offset has been applied (default setting).
- When this function is enabled, the robot moves to the taught position (position offset is not applied). For details of the OFFSET instructions, refer to Sections 6.3.6 and 6.14.

Use Procedure 3-43 to ignore offset.

Procedure 3-43 Setting Ignore Offset

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select General. You will see a screen similar to the following.

SETUP General		JOINT	100 %
			1/4
1 Brake on hold:	DISABLED		
2 Current language:	DEFAULT		
3 Ignore Offset command:	DISABLED		
4 Ignore Tool_offset:	DISABLED		
[TYPE]	ENABLED	DISABLED	

- 5 Move the cursor to Ignore Offset command.
- 6 Enable or disable the Ignore Offset command:
 - To enable this, press F4, ENABLED.
 - To disable this, press F5, DISABLED.

3.16 IGNORE TOOL OFFSET SETUP

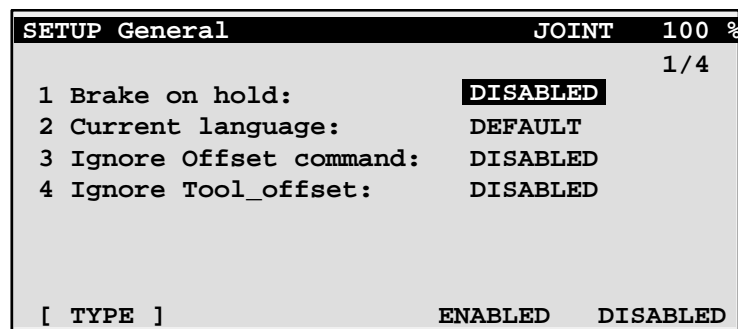
This function ignores the position offset specified by a TOOL OFFSET instruction.

- When this function is disabled, the robot moves to the position for which a tool offset has been applied (default setting).
- When this function is enabled, the robot moves to the taught position (tool offset is not applied). For details of the tool offset instructions, refer to Sections 6.3.6 and 6.15.

Use Procedure 3-44 to set ignore tool offset.

Procedure 3-44 Setting Ignore Tool Offset

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select General. You will see a screen similar to the following.



- 5 Move the cursor to Ignore Tool_offset.
- 6 Enable or disable the Ignore Tool_offset:
 - To enable this, press F4, ENABLED.
 - To disable this, press F5, DISABLED.

3.17 USER ALARM SETUP

The Setting User Alarm screen allows you to define a message that will be displayed on the teach pendant status line. This message is displayed when a user alarm instruction is executed in a teach pendant program.

For example, if you define the message of user alarm 1 (UALM[1]) to be “Perform repair procedure,” and the instruction UALM[1] is executed in a teach pendant program, then the message will be displayed on the status line of the teach pendant as:

INTP-213 Perform repair procedure (name, line) UALM[1]

where *name* is the name of the current program and *line* is the line number in which the UALM[1] instruction was executed. Refer to Section 6.12.2 for more information on the user alarm instruction. Use Procedure 3-45 to set the user alarm.

User Alarm Severity

By default, the severity of a user alarm is STOP, which pauses the program and stops robot motion. If you want to change the severity of the user alarm, you must set the appropriate \$UALRM_SEV[n] system variable to a value that corresponds to the severity you want. “n” corresponds to the number of the user alarm. Refer to Table 3-16 for the actions associated with \$UALRM_SEV[n] values. Use Procedure 3-46 to set user alarm severity.

Table 3-16. \$UALRM_SEV[n] Severity Values

Value	Action
0	No action
2	Pause program
3	Abort program with error
4	Stop program motion
6	Pause program and stop its motion
8	Cancel program motion
10	Pause program and cancel its motion
11	Abort program and cancel its motion

16 added to any value causes servomotors to be turned off.

32 added to any value causes the action to apply to all programs and all motions.

64 added to any value requires a cold start to reset the controller.

For example,

- A value of 0 causes a warning message to be displayed.
- A value of 6 pauses the program and stops its motion.
- A value of 43 aborts all programs and cancels all motions (11 + 32)

Procedure 3-45 Setting User Alarm

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select User Alarm. You will see a screen similar to the following.

Setting/User Alarm		JOINT 10%
Alarm No.	User Message	1/10
[1]:	[██████████]	
[2]:	[]	
[3]:	[]	
[4]:	[]	
[5]:	[]	
[6]:	[]	
[7]:	[]	
[8]:	[]	
[9]:	[]	
[TYPE]		

- 5 Move the cursor to the message you want to set and press ENTER.
- 6 To make the message:
 - a Select a method of naming the message.
 - b Press the appropriate function keys to add the message. The alarm message can contain up to 29 characters. The amount of alarm message displayed will vary depending on the number of characters in the program name.
 - c When you are finished, press ENTER.
- 7 **If you want to set the severity**, perform Procedure 3-46 .
- 8 Add the corresponding user alarm instruction to the program. Refer to Section 6.12.2. The alarm and message will be displayed when the instruction is executed in test cycle or production.

Procedure 3-46 Setting User Alarm Severity

- Step**
- 1 Press MENUS.
 - 2 Select SYSTEM.
 - 3 Press F1, [TYPE].
 - 4 Select Variables. You will see a screen similar to the following.

SYSTEM Variables		JOINT 50%
		1/129
1	\$ANGTOL	[9] of REAL
2	\$APPLICATION	[3] of STRING [21]
3	\$AP_MAXAX	0
4	\$AP_PLUGGED	2
5	\$AP_TOTALAX	16777216
6	\$AP_USENUM	[32] of BYTE
7	\$ASCII_SAVE	FALSE
8	\$AUTOINIT	2
9	\$BLT	0
10	\$CHECKCONFIG	FALSE

[TYPE]

\$UALRM_SEV [10] of BYTE

1	[1]	6
2	[2]	6
3	[3]	6
4	[4]	6
5	[5]	6
6	[6]	6
7	[7]	6
8	[8]	6
9	[9]	6
10	[10]	6

- 5 Move the cursor to \$UALRM_SEV and press ENTER.
- 6 Move the cursor to the number that corresponds to the number of the user alarm for which you want to set the severity.
- 7 Type the number that corresponds to the severity you want and press ENTER. Refer to Table 3-17 for a list of values.

Table 3-17. \$UALRM_SEV[n] Severity Values

Value	Action
0	No action
2	Pause program
3	Abort program with error
4	Stop program motion
6	Pause program and stop its motion
8	Cancel program motion
10	Pause program and cancel its motion
11	Abort program and cancel its motion

16 added to any value causes servomotors to be turned off.

32 added to any value causes the action to apply to all programs and all motions.

64 added to any value requires a cold start to reset the controller.

3.18 OVERRIDE SELECT SETUP

Override select setup allows you to specify four different speed limiting percentages for production operation. When enabled, override select is in effect when the teach pendant is disabled and the REMOTE/LOCAL keyswitch is set to REMOTE.

Using Override Select

You specify two digital inputs to control override select. The four combinations of the values of these digital inputs (ON ON, ON OFF, OFF OFF, OFF ON) correspond to four override percentages. To use override select, you must

1. Define the digital input signals that will be used.
2. Specify the override percentage that corresponds to each of the four digital input value combinations.
3. Enable or activate the override select function when you are ready to use it.

Effect of Override Select

When override select is enabled, the following occurs:

- The jog speed keys on the teach pendant are practically disabled. When you use these keys to change the speed value, the value is quickly changed back to the value set by override select.
- The override instruction has no effect on the speed value set by override select.
- You cannot change the settings of the digital input signal number and override. If you want to change these settings, disable override select in advance.
- If override select is enabled when controller power is turned off, when the controller is turned on again, the speed will return to the value set by override select.
- It is possible to specify the same number as two digital input signal numbers. In this case, only the combinations ON-ON and OFF-OFF are meaningful.
- If override select is disabled by setting the REMOTE/LOCAL keyswitch to LOCAL, the speed stays at the override select value until it is changed by the teach pendant jog speed keys or the override instruction.

Override Select Setup

Table 3-18 lists and describes the items on the override select screen you must set.

Table 3-18. Override Select Menu Listing

ITEM	DESCRIPTION
Function Enable	This item allows you to specify whether or not the override select will function. When set to ENABLE , the override select will limit the speed of the robot during production operation. When set to DISABLE , the speed of the robot will not be clamped during production operation. You must set Function Enable to DISABLE before the Override select percentage can be changed.
Signal 1 DI	This item allows you to specify the digital input (DI) signal whose status (ON or OFF) determines, along with the the status of Signal 2 DI, which of the four override selects are used to clamp the speed of the robot during production operation.
Signal 2 DI	This item allows you to specify the digital input (DI) signal whose status (ON or OFF) determines, along with the the status of Signal 1 DI, which of the four override selects are used to clamp the speed of the robot during production operation.
Signal 1 ON/OFF	This item indicates how the status of the two digital inputs is used to determine which of the four override selects are used to clamp the speed of the robot during production operation.
Signal 2 ON/OFF	This item indicates how the status of the two digital inputs is used to determine which of the four override selects are used to clamp the speed of the robot during production operation.
Override	This item allows you to enter the override select percentage. You must set Function Enable to DISABLE before the override select percentage can be changed.

Use Procedure 3-47 to set up the override select.

Procedure 3-47 Setting Up Override Select

Condition ■ You have set up the digital input signals you want to use for override select.

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select OVRD Select. You will see a screen similar to the following.

OVERRIDE SELECT		JOINT	100 %
			1/7
1	Function Enable:	DISABLE	
2	Signal 1:	SDI[1]	[ON]
3	Signal 2:	SDI[32]	[OFF]
	Signal 1	Signal 2	Override
4	OFF	OFF	10%
5	OFF	ON	10%
6	ON	OFF	10%
7	ON	ON	10%
[TYPE]		ENABLE	DISABLE

- 5 Make sure Function Enable is set to DISABLE. If it is set to ENABLE, move the cursor to Function Enable and press F5, DISABLE.
- 6 To specify the digital Signal 1 or Signal 2, move the cursor to Signal or Signal 2 and enter the digital input number.
- 7 To specify the override select percentage, move the cursor to the override percentage of each combination and enter the new number.
- 8 To enable the override select function, move the cursor to Function Enable and press F4, ENABLE.

3.19 PASSWORD SETUP

A *password* is a combination of up to 12 letters, numbers, and symbols, used to allow authorized personnel access to various operations and screens. The password feature is an option and might not be used at your site. Password protection is inactive unless the password option is installed and the Install user is defined.

Four password *levels* provide access to specific operations and menus. Table 3-19 summarizes the four levels of password authorization.

Table 3-21 in Section 3.19.4 for a detailed description of the screens and operations for each level.

Table 3-19. Password Levels

Level	Operations
Install	Assigns user names, passwords, and levels Clears usernames and passwords Disables and enables the Password Log Sets the number of Password users in the system Can perform all Setup, Program, and Operator operations Note: There can only be one install user.
Setup	Performs operations typically used to set up your system. Refer to Section 3.19.4.
Program	Performs more advanced operations. Refer to Section 3.19.4.
Operator	Performs basic operations. Refer to Section 3.19.4.

CAUTION

If you do not know the Install password, you will be unable to perform several functions. Contact your FANUC Robotics technical representative if you lose or forget your Install password.

Password Operations

If you want to use passwords, you must first identify the Install User for your site. The Install user must assign the Install username and password and then *log in*. After logging in, the Install user *assigns* usernames, levels, and passwords for each user.

NOTE No passwords can be used until the Install username and password are assigned.

After the Install User assigns your username, password level, and password, you must *log in* to work at your assigned level. When you log in, you select your username and type your password. Only one user can be logged in at a time.

When you are finished working, you should *log out*. If you do not log out, the system will *timeout* in the number of minutes specified as the Default User Timeout. After the Default User Timeout expires, or you log out, the system reverts to the Operator level and other users can log in. If you forget to log out, other users can log you out.

If Log events is set to ENABLE by the Install User on the SETUP Passwords screen, password information is logged on the ALARM screen. The Password Log contains information about changes to important data, which user made the changes, and when the changes were made. Refer to Procedure 3-53 .

If you are the Install User, refer to Section 3.19.1 for information on assigning usernames, password levels and passwords. If you are an Operator, Program or Setup User, refer to Section 3.19.2.

NOTE For North American HandlingTool, when you exit (log out) of the password screens, the menuing system will default to the QUICK menus. An option called “Quick/Full Password” is provided. When this option is loaded, you must run Setup Application and you will be prompted to change the default to be FULL menus.

3.19.1 Install User Password Operations

The Install User must:

- Assign the Install username and password (Procedure 3-48)
- Assign usernames, levels, and passwords for all other users (Procedure 3-48)
- Enable, disable, and display the Password Log (Procedure 3-52 and Procedure 3-53 in Section 3.19.3)

Use Procedure 3-48 to assign password levels.

Procedure 3-48 Assigning Usernames and Default Passwords for each Password Level

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select Passwords. You will see a screen similar to the following.

SETUP Passwords		WORLD	VFINE
Current user:		None	
Current level:		OPERATOR	
Default user timeout:		0 min	
Timeout occurs in:		0 min	
Log events:		DISABLE	
Number of users:		10	
[TYPE] USERS LOGOUT PASSWRD HELP			

- 5 Press F2, USERS. You will see a screen similar to the following.

SETUP Passwords		WORLD	VFINE
			1/1
USERNAME	PWD	LEVEL	TIME(min)
1 █	*	INSTALL	0
[TYPE] LOGIN LOGOUT HELP >			
CLEAR		CLR_ALL	HELP >

NOTE The Install username and password **must** be set up first.

Assign Install Username and Password

- 6 Press ENTER. Use the arrow and function keys to type the Install username. When you are finished, press ENTER. You will see a screen similar to the following.

```

1 Uppercase
2 Lower Case
3 Punctuation
4 Options                --Insert--
SETUP Passwords
--Set password for BOB

Old password: '      '
New password: '█    '
Verification: '      '

Old Value:
ABCDEF  GHIJKL  MNOPQR  STUVWX  YZ @*.
```

NOTE The password must contain at least three characters.

CAUTION

Make a written note of the Install password. If you do not know the Install password, you will be unable to perform several functions. Contact your FANUC Robotics technical representative if you lose or forget your Install password.

- 7 Type the new password and press ENTER.
- 8 Type the new password again to verify that the first one is correct and press ENTER. You will see a screen similar to the following.

```

Would you like to be logged in?[YES]
                                     YES      NO
```

Log In

- 9 **If you want to log in** press F4, YES. If you do not want to log in press F5, NO.

NOTE You must log in as the Install User to enter other users.

If you press F4, YES, you will see a screen similar to the following.

If you are logged in, the @ will be displayed to indicate the current username.

SETUP Passwords				WORLD	VFINE
					1/10
	USERNAME	PWD	LEVEL	TIME(min)	
1	@BOB	*	INSTALL	15	
2		*		0	
3		*		0	
4		*		0	
5		*		0	
6		*		0	
7		*		0	
8		*		0	
9		*		0	
Password has been set					
[TYPE]	LOGIN	LOGOUT		HELP >
		CLEAR	CLR_ALL		HELP >

Assign Usernames, Passwords, and Levels

- 10** To assign the next username, move the cursor to the next available username, press ENTER, and use the function keys to enter the username.
 - 11** Move the cursor to PWD, press ENTER, and use the function keys to enter the password.
 - 12** Move the cursor to LEVEL, press F4, [CHOICE], and select a level.
 - 13** Move the cursor TIME and type a Default User Timeout value. You can adjust the Default User Timeout value from 0 to 10080 minutes (seven days).
- NOTE** If the Default User Timeout value is 0 when you log in, a timeout will not occur.
- 14** Repeat Steps 10 through 13 for each user you want to have access to the system.
 - 15** To clear the current username and password, press NEXT, >, and then press F2, CLEAR.
 - 16** To clear all usernames and passwords for all users except the Install user, press NEXT, >, and then press F3, CLR_ALL.

17 To modify the number of usernames in the system,**⚠ CAUTION**

If you modify the number of usernames to be fewer than the number of users currently assigned, some users will be deleted from the system.

- a** Press PREV to display the first SETUP Passwords screen.
- b** Move the cursor to Number of users and press ENTER. You can set the number of users to a minimum of 10 and a maximum of 100.

If you are increasing the number of users, you will see the following prompt.

```
Enter number of users for passwords:
```

- c** Type the new number of users and press ENTER. You will see a screen similar to the following.

```
Changing number of users.
```

If you want to decrease the number of users, you will see the following prompt.

```
Reconfiguring.  DELETE users?[NO]
                YES      NO
```

To delete the users press F4, YES. To cancel the operation press F5, NO.

- d** Turn off the controller then turn it on again to accept the new list of users.

Log Out**18 To log out** press F3, LOGOUT.

NOTE After the Default User Timeout expires, or you log out or turn off the controller, the system reverts to the Operator level.

3.19.2 Program and Setup User Password Operations

Program and Setup users can:

- Log in (Procedure 3-49)
- Log out (Procedure 3-50)
- Change their password (Procedure 3-51)
- Display the Password Log (Procedure 3-53 in Section 3.19.3)

Procedure 3-49 Logging In

- Condition**
- Passwords have already been set up. (Refer to Section 3.19.1)
 - No user is currently logged in. Only one user can be logged in at a time.

NOTE If you do not know your username and password, contact the Install User.

- Step**
- 1 If you are using FULL menus,**
 - a Press MENUS.
 - b Select SETUP.
 - c Press F1, [TYPE].
 - d Select Passwords.
 - 2 If you are using QUICK menus,**
 - a Press MENUS.
 - b Select SETUP PASSWORDS.
 - 3** Press F2, USERS.
 - 4** Move the cursor to your username.
 - 5 To log in,** press F2, LOGIN. You will see a screen similar to the following.

```

1 Uppercase
2 Lower Case
3 Punctuation
4 Options                                --Insert--
SETUP Passwords
--Password for MARY

Enter password: '          '

Old Value:
ABCDEF  GHIJKL  MNOPQR  STUVWX  YZ @*.
```

- 6** Type your password and press ENTER.

7 If you want to change the timeout value, move the cursor to the TIMEOUT value for the current user and type a new timeout value. You can adjust the Default User Timeout value from 0 to 10080 minutes (seven days).

NOTE If the Default User Timeout value is 0 when you log in, the timeout will not occur.

NOTE Only one user can be logged in at a time. If another user is currently logged in, you must choose whether or not to log them out before you can log in. You will see the following prompt.

```
User JACK logged in. Force logout?[NO]
                                     YES   NO
```

8 To log out the current user press F4, YES. Otherwise press F5, NO.

If you select F4, YES, you will see a screen similar to the following.

If you are logged in, the @ will be displayed to indicate the current username.

SETUP Passwords			WORLD	VFINE
				1/1
	USERNAME	PWD	LEVEL	TIME(min)
1	JACK	*	INSTALL	0
2	@MARY	-	SETUP	15
3		*		0
4		*		0
5		*		0
6		*		0
7		*		0
8		*		0
9		*		0
[TYPE]	LOGIN	LOGOUT		HELP
[TYPE]	CLEAR	CLR_ALL		HELP

Procedure 3-50 Logging Out

Condition ■ Passwords have already been set up. (Refer to Section 3.19.1)

■ You are currently logged in. (Refer to Procedure 3-49)

Step 1 If you are using FULL menus,

- a Press MENUS.
- b Select SETUP.
- c Press F1, [TYPE].
- d Select Passwords.

2 If you are using QUICK menus,

- a Press MENUS.
- b Select SETUP PASSWORDS.

3 To log out, press F3, LOGOUT. After you log out the system reverts to the Operator level.

NOTE When you log out, time out, or are an Operator user, the QUICK menu will be displayed.

Procedure 3-51 Changing Your Password

- Condition**
- Passwords have already been set up. (Refer to Section 3.19.1)
 - You are currently logged in. (Refer to Procedure 3-49)
- Step**
- 1 If you are using FULL menus,**
 - a Press MENUS.
 - b Select SETUP.
 - c Press F1, [TYPE].
 - d Select Passwords.
 - 2 If you are using QUICK menus,**
 - a Press MENUS.
 - b Select SETUP PASSWORDS. You will see a screen similar to the following.

```

SETUP Passwords          WORLD  VFINE
-----
Current user:           AAAA
Current level:         INSTALL
1 Default user timeout: 15 min
2 Timeout occurs in:   4 min
3 Log events:          DISABLE
4 Number of users:     10

[ TYPE ]  USERS  LOGOUT  PASSWRD  HELP

```

- 3** Press F4, PASSWRD. You will see a screen similar to the following.

```

1 Uppercase
2 Lower Case
3 Punctuation
4 Options          --Insert--
SETUP Passwords
--Set password for AAAA

Old password: '      '
New password: '      '
Verification: '      '

Old Value:
ABCDEF  GHIJKL  MNOPQR  STUVWX  YZ_@*.
```

- 4** Type the old password and press ENTER.
- 5** Type the new password and press ENTER.
- 6** Type the new password again to verify the first one is correct and press ENTER. The new password is set.

3.19.3 Password Log

If the Log Events item is set to ENABLE by the Install user on the SETUP Passwords screen, the following events will be displayed in the Password Log:

- Password events
- Programming events
- File manipulation events

Each time an event occurs, such as when a user logs in or when a program is created, the event is logged in the Password Log.

Only the Install user can enable the Logs Events item. Use Procedure 3-52 to enable the Password Log.

Any user can display the Password Log. Use Procedure 3-53 to display the Password Log.

Refer to Table 3-20 for a listing of the password error messages (PWD).

Table 3-20. Password Error Messages

Message	Description
Password Events	
PWD-001 Login (%s) Install	The specified user logged in at the Install level.
PWD-002 Logout (%s) Install	The specified user logged out from the Install level.
PWD-003 Login (%s) Setup	The specified user logged in at the Setup level.
PWD-004 Logout (%s) Setup	The specified user logged out from the Setup level.
PWD-005 Login (%s) Program	The specified user logged in at the Program level.
PWD-006 Logout (%s) Program	The specified user logged out from the Program level.
PWD-007 Password Timeout (%s)	The specified user's timeout expired.
PWD-031 QUICK MENUS forced	QUICK menus have been displayed.
Programming Events	
PWD-008 Create Program %s.TP	The specified program has been created.
PWD-009 Delete program %s.TP	The specified program has been deleted.
PWD-010 Rename %s.TP %s.TP	The specified program has been renamed to the name specified.
PWD-011 Set %s.TP subtype from %s to %s	The subtype of the specified program has been changed.
PWD-012 Set %s.TP comment	The comment of the specified program has been changed.
PWD-013 Set %s.TP group mask	The group mask of the specified program has been changed.
PWD-014 Set %s.TP write protect on	Write protection for the specified program has been set to on.
PWD-015 Set %s.TP write protect off	Write protection to the specified program has been set to off.
PWD-016 Set %s.TP ignore pause on	Ignore pause for the specified program has been set to on.
PWD-017 Set %s.TP ignore pause off	Ignore pause for the specified program has been set to off.
PWD-018 Write line %d, %s.TP	The specified line has been added to the specified program.
PWD-019 Delete line %d, %s.TP	The specified line has been deleted from the specified program.
PWD-020 Write pos %d, %s.TP	The specified position has been added to the specified program.

Table 3-20. (Cont'd) Password Error Messages

Message	Description
PWD-021 Delete pos %d, %s.TP	The specified position has been deleted from the specified program.
PWD-022 Renumber pos %d as %d, %s.TP	The specified position has been renumbered to the specified position, in the specified program.
PWD-023 Set application data %s.TP	Application data has been set in the specified program.
PWD-024 Delete application data %s.TP	Application data has been deleted from the specified program.
File Manipulation Events	
PWD-025 Load %s	The specified file has been loaded.
PWD-026 Load %s as Program %s	The specified file has been loaded as the specified program.

Procedure 3-52 Enabling the Password Log

- Condition** ■ You are logged in as the Install User. (Procedure 3-48)
- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select Passwords. You will see a screen similar to the following.

SETUP Passwords		WORLD	VFINE
			3/4
Current user:		AAAA	
Current level:		INSTALL	
1 Default user timeout:		15 min	
2 Timeout occurs in:		4 min	
3 Log events:		DISABLE	
4 Number of users:		10	
[TYPE] USERS LOGOUT ENABLE DISABLE			

- 5 To disable or enable the Password Log,
 - a Move the cursor to Log events.
 - b To enable log events, press F4, ENABLE.
 - c To disable log events, press F5, DISABLE.

Procedure 3-53 Displaying the Password Log

- Condition**
- The Install User has set Log events to ENABLE. (Procedure 3-52)
 - You are logged in at the Install, Program, or Setup level.
 - FULL menus are displayed.

- Step**
- 1 Press MENUS.
 - 2 Select ALARM.
 - 3 Press F1, [TYPE].
 - 4 Select Password Log. You will see a screen similar to the following.

```

TEST1                LINE 15                ABORTED
Alarm: HIST                WORLD 100 %
                            1/100
 1 PWD -001 Login   (BOB) Install
 2 PWD -002 Logout  (BOB) Install
 3 PWD -001 Login   (MARY) Setup
 4 PWD -007 Password Timeout (MARY)

[ TYPE ]                ACTIVE  CLEAR  HELP

```

Refer to Table 3-20 for a listing of the PWD messages.

NOTE Refer to Appendix A for more information on the PWD error messages.

3.19.4 Password Level Screen Permissions

Depending on which level you are logged in, the following password screen permissions are available:

- n/a = The screen is not displayed
- C = The information on the screen can be displayed, changed and operations can be performed
- D = The screen can only be displayed (you cannot change any information on the screen)

Table 3-21 lists each screen and the corresponding password level permissions for each level.

Table 3-21. Password Level Screen Permissions

Menu	Teach Pendant Screen	Password Level				
		Install	Setup	Program	Full Operator	Quick Operator
Utilities	Hints	D	D	D	D	D
	Program Adjust	C	C	D	D	n/a
Test Cycle	Test Run	C	C	C	C	C
Manual Functions	Macro Manual Functions	C	C	C	C	C
	Error Recovery	C	C	C	C	n/a
Alarm	Alarm	C	D	D	D	D
	Password Log	C	D	D	D	D
I/O	Digital	C	C	D	D	D
	Analog	C	C	D	D	n/a
	Group	C	C	D	D	D
	Robot	C	C	D	D	D
	UOP	C	C	D	D	n/a
	SOP	C	C	D	D	n/a
	Inter Connect	C	C	D	D	n/a
	I/O Link Device	C	C	D	D	n/a
	PLC I/O	C	C	D	D	n/a
	Load Clutch	C	C	C	C	n/a

n/a = Screen not available

C = You can display, change, and perform operations on the screen

D = You can only display the screen

Table 3-21. (Cont'd) Password Level Screen Permissions

Menu	Teach Pendant Screen	Password Level				
		Install	Setup	Program	Full Operator	Quick Operator
Setup	General Setup	C	C	D	D	n/a
	Frame Setup	C	C	D	D	n/a
	Port Init	C	C	D	D	n/a
	Macro	C	C	D	D	n/a
	Reference Position	C	C	D	D	n/a
	User Alarm	C	C	D	D	n/a
	Override Select	C	C	D	D	n/a
	RSR/PNS	C	D	D	D	n/a
	Passwords	C	C	C	C	C
	Host Communications	C	D	D	D	n/a
	Space Check	C	C	C	C	n/a
	Error Recovery	C	C	C	C	n/a
	Group Motion	C	C	C	C	n/a
	Coordinated Motion	C	D	D	D	n/a
	Line Track Setup	C	C	C	C	n/a
	Encoder Setup	C	C	C	C	n/a
	Continuous Turn	C	C	C	C	n/a
Softfloat	C	C	C	C	n/a	
File	File	C	C	C	D	n/a
	File Memory	C	C	C	D	n/a
	Controller Backup (CTRL2 START)	C	C	C	C	n/a
Status	Axis Status	C	D	D	D	D
	Software Version	C	C	C	C	n/a
	Safety Signals	C	C	C	C	n/a
	Display Memory	C	C	C	C	n/a
	Program Timer	C	D	D	D	n/a
	System Timer	C	D	D	D	n/a
	Exec-History	C	D	D	D	n/a
	Conditions	C	D	D	D	n/a
User	User	C	C	C	C	n/a

n/a = Screen not available

C = You can display, change, and perform operations on the screen

D = You can only display the screen

3. GENERAL SETUP

Table 3-21. (Cont'd) Password Level Screen Permissions

Menu	Teach Pendant Screen	Password Level				
		Install	Setup	Program	Full Operator	Quick Operator
Select	Select	C	C	C	C	n/a
Edit	Edit	C	C	C	C	n/a
Data	Register	C	C	C	D	D
	Position Register	C	C	C	D	D
	Load Clutch	C	C	C	C	n/a
	Palletizing Register	C	C	C	C	n/a
Position	Position	C	C	C	C	C
System	System Variables	C	C	C	D	n/a
	Servo Parameters	D, C	D	D	D	n/a
	Master/Calibrate	C	D	D	D	n/a
	Axis Limits	C	C	D	D	n/a
	Clock	C	C	D	D	n/a
	Overtravel Release	C	C	C	C	n/a
	System Config	C	D	D	D	n/a
Tools	Tool1	C	C	C	C	C
	Tool2	C	C	C	C	C
Vision	Vision	C	C	C	C	n/a

n/a = Screen not available
 C = You can display, change, and perform operations on the screen
 D = You can only display the screen

3.20 ROBOT PAYLOAD SETTING

Robot payload is the weight of the robot end-of-arm tooling and workpiece. If you have not set up the proper robot payload during software installation, or if you need to change the robot payload because you have changed end-of-arm tooling or the workpiece, you must set robot payload.

NOTE Care should be taken to set payload values as accurately as possible. More accurate values might improve position accuracy and cycle time.

NOTE Automatic robot payload estimation is available for some robot models. If your robot model does not have the payload setting feature, the message, “IDENT is not supported to this robot,” will be displayed when you press F2, IDENT.

You can define up to ten different payload schedules. You can then specify a payload schedule by using the payload setup screens and by using the payload teach pendant program instructions. Refer to Section 3.20.4 for more information on the payload teach pendant program instructions.

3.20.1 Payload Setting Process

When you set payload, you must do the following:

1. Perform payload *calibration*. This means defining the payload of the robot without end-of-arm tooling.
2. Perform payload *estimation*. This means defining the payload of the robot with end-of-arm tooling. You must perform payload estimation **after** you perform payload calibration.

If you do not want to perform payload calibration and estimation, but want to return the payload settings to the default values, you can reset them to the default values.

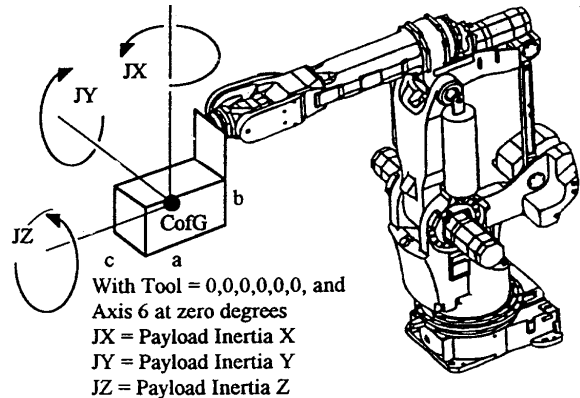
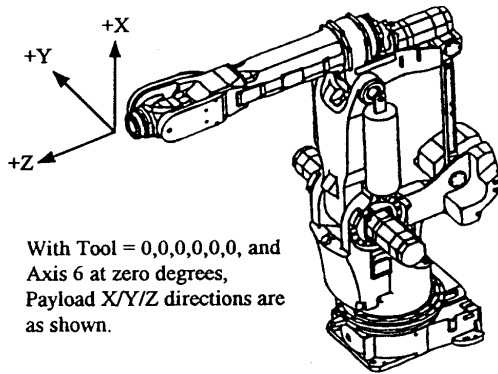
3.20.2 Payload Setting Items

When you set payload, you set the values of several items related to payload. Refer to Table 3-22 for a short description of the items you must set. Direction is relative to the robot tool frame with X, Y, Z, W, P, and R set to zero and robot joint angles at the zero positions.

3. GENERAL SETUP

Table 3-22. SYSTEM Payload Screen

ITEM	DESCRIPTION
Payload (kg)	Weight of the end-of-arm tooling. Note: There are 2.21 pounds in a kilogram.
Payload center X (cm)	The up and down offset of the payload center of gravity from the center of the faceplate. Positive (+) values are up. There are 0.39 inches per centimeter.
Payload center Y (cm)	The side offset of the payload center of gravity from the center of the faceplate. Positive (+) values are to the right of the faceplate when viewed from behind the faceplate. There are 0.39 inches per centimeter.
Payload center Z (cm)	The offset of the payload center of gravity from the center of the faceplate. Positive (+) values are out from the faceplate. There are 0.39 inches per centimeter.
Payload inertia X (kgfcm²)	The moment of inertia of the payload around an axis parallel to the X-direction for the tool frame and through the center of gravity of the payload.
Payload inertia Y (kgfcm²)	The moment of inertia of the payload around an axis parallel to the Y-direction for the tool frame and through the center of gravity of the payload.
Payload inertia Z (kgfcm²)	The moment of inertia of the payload around an axis parallel to the Z-direction for the tool frame and through the center of gravity of the payload.
Arm load axis #1 (kg)	Additional weight mounted to axis 1.
Arm load axis #3 (kg)	Additional weight mounted to axis 3.



In the example shown above, assume a = 60cm, b = 15cm, c = 10cm
 (Material = steel, density = 0.00783 kg/cm³)
 Then M = Mass = 60 x 15 x 10 x 0.00783 = 70kg

From this, using standard Inertia formulae, the Payload Inertias can be calculated:

$$J_x = \frac{M}{12}(a^2 + c^2) = \frac{70}{12}(60^2 + 10^2) = 21583 \text{kg.cm}^2 = 22.0 \text{kgf.cm.s}^2$$

$$J_y = \frac{M}{12}(a^2 + b^2) = \frac{70}{12}(60^2 + 15^2) = 22312 \text{kg.cm}^2 = 22.7 \text{kgf.cm.s}^2$$

$$J_z = \frac{M}{12}(b^2 + c^2) = \frac{70}{12}(15^2 + 10^2) = 1896 \text{kg.cm}^2 = 1.9 \text{kgf.cm.s}^2$$

Note 1: (1kgf.cm.s² = 980kg.cm²)

Note 2: Equation $\left(J_n = \frac{M}{12}(\text{width}^2 + \text{depth}^2) \right)$ only applies to simple load shape shown. Other shape loads will require different inertia calculation formulae.

3.20.3 Payload Setup Procedure

Use Procedure 3-54 to set robot payload.

Procedure 3-54 Setting Robot Payload

NOTE Automatic robot payload estimation is not available for all robot models. If your robot model does not have the payload setting feature, the message, "IDENT is not supported to this robot," will be displayed when you press F2, IDENT.

NOTE You cannot update payload values when a program is running and the active schedule number is the same as the displayed schedule you want to modify.

- Condition**
- SRDY is on.
 - No motion commands have been issued.
 - \$PARAM_GROUP[].\$MOUNT_ANGLE has not been set.
 - Robot mastering/calibration has been performed.
- Step**
- 1 Press MENUS.
 - 2 Select SYSTEM.
 - 3 Press F1, [TYPE].
 - 4 Select Motion. You will see a screen similar to the following.

SYSTEM MOTION PERFORMANCE			JOINT	50%
Group 1				1/10
No.	PAYLOAD[kg]	Comment		
1	120.00	[]
2	120.00	[]
3	120.00	[]
4	120.00	[]
5	120.00	[]
6	120.00	[]
7	120.00	[]
8	120.00	[]
9	120.00	[]
10	120.00	[]
Active PAYLOAD number = 1				
[TYPE]	GROUP	DETAIL	ARMLoad	SETIND >
IDENT				>

Setting Up Payload Information Manually

- 5 To set up payload information manually** for the schedule you chose, move the cursor the payload schedule you want and press F3, DETAIL. You will see a screen similar to the following

SYSTEM MOTION PERFORMANCE		JOINT	50%
Group 1			
1	Schedule No[1]:	[*****]	
2	PAYLOAD	[kg]	120.00
3	PAYLOAD CENTER X	[cm]	30.00
4	PAYLOAD CENTER Y	[cm]	25.00
5	PAYLOAD CENTER Z	[cm]	25.00
6	PAYLOAD INERTIA X	[kgfcms ²]	77.00
7	PAYLOAD INERTIA Y	[kgfcms ²]	306.00
8	PAYLOAD INERTIA Z	[kgfcms ²]	306.00
Please power off/on after modification			
[TYPE]	GROUP	NUMBER	DEFAULT HELP

- a To display help** for the items on the screen, press F5, HELP. To display more information, use the arrow keys. When you are finished displaying help information, press PREV.
 - b** Press F3, NUMBER, and enter the number of the payload schedule for which you want to set up payload information manually.
 - c** Move the cursor to the items you want to set and set them as desired. All x, y, and z dimensions are relative to the faceplate.
- 6 To select a different motion group**, press F2, GROUP, and specify the motion group you want.
- 7 To set payload values to the default values set at FANUC Robotics**, press and hold SHIFT and press F4, DEFAULT and then do one of the following:
- To confirm the change to the factory default values, press F4, YES.
 - To cancel the default settings and return to the previous settings, press F4, NO.

NOTE You must cold start the robot for these changes to take effect (Step 12).

Setting Up Arm Load Information

- 8** To set arm load information, press PREV until the payload schedule listing screen is displayed, and press F4, ARMLOAD. You will see a screen similar to the following.

SYSTEM MOTION PERFORMANCE			JOINT	50%
Group 1				
1	ARM LOAD AXIS #1	[kg]	250.00	
2	ARM LOAD AXIS #3	[kg]	20.00	
Please power off/on after modification				
[TYPE]	GROUP		DEFAULT	HELP

- a** To display help for the items on the screen, press F5, HELP. To display more information, use the arrow keys. When you are finished displaying help information, press PREV.

- b** Move the cursor to the item you want to set and set it as desired.

Payload Calibration

- 9** To perform payload calibration, do the following:

- a** Make sure the end-of-arm tooling is **not attached** to the robot arm.

- b** Jog the robot to a reference position. Define the reference position so that the robot is fully extended. Make sure the robot can reach this position when the end-of-arm tooling is mounted on the robot arm. Use the following ranges of joint angles:

The joint angle of J5 depends on J3.

Since you will need to use this same position during payload estimation, record this position in a motion instruction in a program, or in a position register.

- c** Press PREV until the payload schedule listing screen is displayed.
- d** Press NEXT, >, and then press F2, IDENT. You will see a screen similar to the following.

SYSTEM MOTION PERFORMANCE			JOINT	50%
Group 1				
Schedule No[1]: [*****]				
1	PAYLOAD CALIBRATION		****	
2	PAYLOAD ESTIMATION		****	
PAYLOAD CALIBRATION POSITION				
J1<*****>	J2<*****>	J3<*****>		
J4<*****>	J5<*****>	J6<*****>		
J7<*****>	J8<*****>	J9<*****>		
Estimated payload : ***** kg				
[TYPE]	GROUP	NUMBER	EXECUTE	DELETE

- e** Press F3, NUMBER, and enter the number of the payload schedule for which you want to perform payload calibration.

- f Move the cursor to PAYLOAD CALIBRATION.
- g Press and hold SHIFT and press F4, EXECUTE.
- h Calibration will be performed. When it is finished, the PAYLOAD CALIBRATION status will be changed to DONE and the payload calibration position will be displayed.
- i **To delete calibration data**, press and hold SHIFT and press F5, DELETE.

Payload Estimation 10 Perform the following steps for payload estimation:

- a After you have performed payload calibration (Step 9), attach the end-of-arm tooling to the robot arm.
- b Jog the robot to the reference position you defined during payload calibration. If you defined the reference position using a position register, display the DATA Position Reg screen and move to the position.
- c Move the cursor to PAYLOAD ESTIMATION.
- d Press F3, NUMBER, and enter the number of the payload schedule for which you want to perform payload estimation.
- e Press and hold SHIFT and press F4, EXECUTE.

The payload will be estimated. See the following screen for an example.

```
Estimated payload : 123.45 kg
Path and Cycletime will change. Set it?
                                     YES  NO
```



WARNING

Make sure that the payload schedule you define matches the correct payload information before you continue; otherwise, the robot will not move the way you expect, and could injure personnel or damage equipment.

- f Decide whether to accept the estimated payload:
 - To accept the payload, press F4, YES.
 - To reject the payload, press F5, NO.

NOTE You must cold start the robot for these changes to take effect (Step 12).

-
- Set the Active Payload** **11** To set the active payload
- a** Press PREV until the payload schedule listing screen is displayed.
 - b** Press F5, SETIND.
 - c** Type the number of the payload schedule you want and press ENTER.
- When you are finished** **12** **When you are finished setting payload information**, cold start the robot:
- a** On the teach pendant, press and hold the PREV and NEXT keys.
 - b** While still pressing PREV and NEXT on the teach pendant, press the ON button on the operator panel or operator box.
 - c** After the BMON> prompt appears on the teach pendant screen, release the PREV and NEXT keys.
 - d** Press F1, COLD, and press ENTER.
 - e** Press F5, START, and press ENTER.
- BMON>**
- BMON> COLD**
- BMON> START**

3.20.4 Payload Teach Pendant Program Instruction

Some applications and the Collision Guard function require the proper setting of payload information. If the payload changes during your application, you must use the PAYLOAD[GPx:y] instruction to select the appropriate payload schedule, y, for the specified motion group, x. Refer to Section 6.23 for details on the PAYLOAD[GPx:y] instruction.

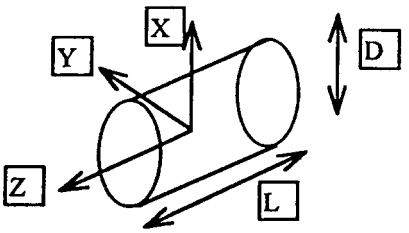
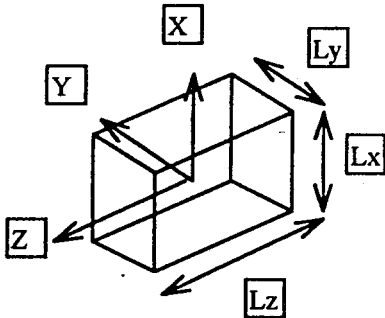
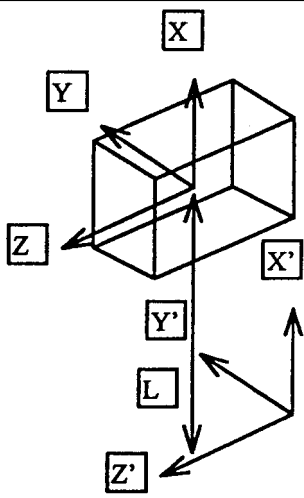
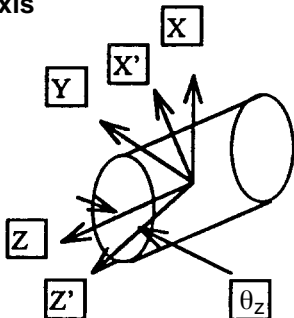
The PAYLOAD[GPx:y] instruction allows you to specify the *payload schedule* to use. You can specify up to 10 different sets of payload information. Each set of payload information corresponds to a schedule number.

Before you use a PAYLOAD[GPx:y] instruction, you must make sure you have set up the payload schedule that corresponds to the one you specify. Refer to Section 3.20 for information on setting up payloads.

3.20.5 Inertia Equations

Refer to Figure 3-41 for inertia equations to use in calculating inertia.

Figure 3-41. Inertia Equations

<p>Cylinder M = Mass, D= Diameter, L = Length, ρ = Density Equation 1</p> $M = \left(\pi \frac{D^2}{4} L \right) \rho$ <p>Inertia about own C of G parallel to X, Y, Z axes Equation 2</p> $J_x = J_y = M \left(\frac{L^2}{3} + \frac{D^2}{16} \right)$	<p>Equation 3</p> $J_z = M \left(\frac{D^2}{8} \right)$	
<p>Cuboid Equation 4</p> $M = (L_x L_y L_z) \rho$ <p>Inertia about own C of G parallel to X, Y, Z axes Equation 5</p> $J_x = \frac{M}{12} (L_y^2 + L_z^2)$ <p>Equation 6</p> $J_y = \frac{M}{12} (L_x^2 + L_z^2)$ <p>Equation 7</p> $J_z = \frac{M}{12} (L_x^2 + L_y^2)$		
<p>Inertia of Object about Axis Parallel to Major Axis Inertia about axis Z through own C of G = Jz Inertia about axis Z', parallel to Z' at distance L = J'z</p> <p>Equation 8</p> $J'_z = J_z + ML^2$		
<p>Inertia of Object about Axis at Angle to Major Axis Equation 9</p> $J'_x = J_x \cos^2 \theta_x + J_y \cos^2 \theta_y + J_z \cos^2 \theta_z$		

4 SYSTEM CONFIGURATION SETUP

4

SYSTEM CONFIGURATION SETUP

Topics In This Chapter		Page
System Configuration Setup Screen	This section contains descriptions of each of the items on the System Configuration Setup screen.	4-2
System Configuration Setup Procedure	This section contains a procedure for performing system configuration setup. . .	4-6

The system configuration menu contains items that must be set when the system is installed.

The following items can be set or referenced using the System Config screen:

- Power failure handling enable and disable
- Program automatically activated at power on
- Completion signal for power-failure handling
- Cell I/O signals enable and disable
- External start signal set to resume programs
- CSTOPI input
- PROD_START input enabled with confirmation signal
- Detection of the RESET input at rising and falling edge
- Abnormal pressure detection enable and disable
- Time limit for wait instructions
- Time limit for receive instructions
- Program search enable and disable after a program has been finished
- Words registered as program names
- Default logical instruction setting
- Upper and lower limits for ACC instruction
- Batched addition and deletion using Wjnt
- Aborting all programs with CSTOPI input

4.1 SYSTEM CONFIGURATION SETUP SCREEN

Table 4-1 contains the screen items from the System Configuration Setting screen, a detailed description of each item, and any related system variable information.

Table 4-1. System Configuration Setup Screen Items

Screen Item	Description	Related System Variable
Use HOT START	When power-failure handling is enabled, hot start is performed upon power on.	\$SEMIPOWERFL
I/O power fail recovery	This item specifies how the recovery of I/O is performed when hot start is valid. This also specifies how the recovery of simulated I/O is performed when hot start is invalid. There are four modes in hot start:	\$PWF_IO
NOT RECOVER	Recovery of Input/Output signals cannot be performed regardless of the setting for hot start. All outputs are turned off and simulated mode is released.	
RECOVER SIM	Simulated mode is recovered regardless of the setting for hot start. However, all real output signals and simulated Input/Output signals are turned off.	
UNSIMULATE	Output signals are recovered when hot start is valid, but all simulated mode is released. Since output signals are not recovered when hot start is invalid, this causes the same result as NOT RECOVER.	
RECOVER ALL	Input/Output signals are recovered when hot start is valid. Output signals and simulated mode are in the same state as they were at power down. Since output signals are not recovered when hot start is invalid, this causes the same result as RECOVER SIM.	
Autoexec program for Cold start	This item specifies the program name that is executed automatically when you turn on the controller and HOT START is invalid. The specified program is executed just after turning on the controller.	\$PWR_NORMAL
Autoexec program for Hot start	This item specifies the name of the program that is activated automatically upon power on when power-failure handling is enabled. The specified program is executed immediately after the power is turned on. If the specified program is not executed after 15 seconds have elapsed, this program is aborted. Note: Since the program activated automatically at power on is executed before the power is applied to the servo, that program cannot operate the robot. Therefore, specify only a program designed specifically for system setup or for initializing I/O devices. Also, specify the attributes for the program on the detail program screen, as follows: Group Mask : [*,*,*,*,*] Ignore pause : [TRUE]	\$PWR_SEMI
HOT START done signal	This completion signal for power-failure handling specifies the digital output signal (DO), issued when power-failure handling (hot start) has been completed. When power-failure handling is not done, this signal is off. When 0 is specified for this signal, this signal is not output.	\$SEMIPWFDO

4. SYSTEM CONFIGURATION SETUP

Table 4-1. (Cont'd) System Configuration Setup Screen Items

Screen Item	Description	Related System Variable
Restore selected program	This item specifies whether the program that was selected when the power was cut after a cold start, is selected again after the power is turned back on. When this item is enabled, the program selected when the power was cut is selected when the power is turned back on. When this item is disabled, no program is selected when the power is turned on again.	\$DEFPROG_ENB
Enable UI signals	UOP I/O signals are enabled or disabled. When disabled, peripheral unit input signals (UI[1] to UI[18]) are ignored. Refer to Section 3.3 for more information on UOP signals.	\$OPWORK.\$uop_disable
START for CONTINUE only	When this external start signal is set to enable (TRUE), the external activation signal (START) activates only the suspended programs. Refer to Section 3.3 for more information on UOP signals.	\$SHELL_CFG.\$cont_only
CSTOPI for ABORT	When CSTOPI forces a program to terminate, program termination by CSTOPI is enabled, the CSTOPI input immediately terminates the program that is currently being forcibly executed. Refer to Section 3.3 for more information on UOP signals.	\$SHELL_CFG.\$use_abort
Abort all programs by CSTOPI	<p>This item selects whether the CSTOPI signal aborts all programs in a multi-tasking environment. When TRUE is specified for this item, the CSTOPI input signal functions as follows:</p> <ul style="list-style-type: none"> -Aborts all programs if RSR is selected for RSR/PNS. -Aborts the selected program if PNS is selected for RSR/PNS. If no program is selected, however, all programs are aborted. <div style="text-align: center;"> <pre> graph TD Start([Start]) --> Q1{Is "CSTOPI for ABORT" TRUE?} Q1 -- no --> A1[Abort selected program] Q1 -- yes --> Q2{RSR / PNS?} Q2 -- RSR --> A2[Abort all programs] Q2 -- PNS --> Q3{Is any program selected?} Q3 -- yes --> A3[Abort selected program] Q3 -- no --> A4[Abort all programs] A1 --> End([End]) A2 --> End A3 --> End A4 --> End </pre> </div> <p>When FALSE is specified for "Abort all programs by CSTOPI," the CSTOPI input signal aborts only a selected program (in the same way as the conventional specifications). When using the CSTOPI input signal in the same way as for a conventional system, specify FALSE (default setting).</p>	\$SHELL_CFG.\$cstopi_all
PROD_START depends on PNSTROBE	When the PROD_START input (enabled with the confirmation signal) is enabled, the PROD_START input is effective only when the PNSTROBE input is on. Enabling this item prevents the program indicated on the teach pendant, which is not to be activated, from being activated erroneously by noise or an incorrect sequence.	\$SHELL_CFG.\$prodstartyp
Detect FAULT_RESET signal	The detection of RESET input at a rising or falling edge specifies whether the signal is detected at a rising or falling edge. Once this item has been set or modified, turn off the power, then turn it on again to enable the setting. Cold start is automatically performed for this operation.	\$SCR.\$resetinvert

Table 4-1. (Cont'd) System Configuration Setup Screen Items

Screen Item	Description	Related System Variable
Use PPABN signal	Abnormal-pressure (*PPABN) detection enable or disable is specified for each motion group. Position the cursor on this item and press the ENTER key to display the screen for setting detection enable or disable for each group. When the *PPABN signal is not used, set this item to disable. After this item has been set or modified, turn off the power, then turn it on again to enable the setting. Cold start is automatically performed for this operation.	\$PARAM_GROUP .\$ppabn_enbl
WAIT timeout	This item specifies the time limit for conditional wait instructions, WAIT..., Timeout, LBL[...]	\$WAITTMOUT
RECEIVE timeout	This item specifies the time limit for register receive instructions, RCV R[...], Timeout, LBL[...]. The time limit can be taught only when the optional sensor interface is specified.	\$RCVTMOUT
Return to top of program	Program search enable or disable after a program has terminated specifies whether the cursor is positioned at the start of the program upon termination of that program. When this item is enabled, the cursor remains positioned at the end of the program (not positioned at the start of the program) upon termination of the program.	\$PNS_END_CUR
Original program name	This item specifies the words displayed for the soft keys on the program creation screen. It is convenient to specify words that are frequently used for program names.	\$PGINP_WORD[1] to [5]
Default logical command	Positioning the cursor to this item and pressing the ENTER key displays the screen for setting the default logical instruction function keys.	
Function (page-key) Name	This item specifies the words, using up to seven characters, displayed for the function keys.	\$DEFLOGOP[*].\$func_title, \$DEFLOGIC[*].\$func_title).
Lines	This item specifies the number of logic statements assigned to one function key. Up to four default logic statements can be assigned to each function key. When 0 is specified, the default logic instruction teaching function is disabled.	\$DEFLOGOP[*].\$total_num, \$DEFLOGIC[*].\$total_num
Maximum of ACC instruction	This item specifies the maximum for the override specified with an acceleration or deceleration override instruction, ACC.	\$ACC_MINLMT
Minimum of ACC instruction	This item specifies the minimum for the override specified with an acceleration or deceleration override instruction, ACC.	
WJNT for default motion	This function adds a Wjnt motion option to the default motion instruction such as linear or circular, or deletes it at the same time. –When F4, [ADD], is pressed, the Wjnt additional instruction is added to every default linear or circular motion instruction. Then, the display on the system configuration screen switches from DELETE (or *****) to ADD. In the prompt line, “Added WJNT to default motion” appears. –When F5, [DELETE], is pressed, the Wjnt additional instruction is deleted from every default linear or circular motion instruction. Then, the display on the system configuration screen switches from ADD (or *****) to DELETE. In the prompt line, “Deleted WJNT from default motion” appears.	

4. SYSTEM CONFIGURATION SETUP

Table 4-1. (Cont'd) System Configuration Setup Screen Items

Screen Item	Description	Related System Variable
Auto display of alarm menu	This item selects whether the alarm screen is displayed automatically. When TRUE is specified for this item, the alarm screen is displayed automatically. The default setting is FALSE. When this item has been set or modified, turn off the power, then turn it on again to enable the setting.	\$ER_AUTO_ENB
Force Message	When the teach pendant MESSAGE instruction is executed, the USER screen is displayed automatically, by default (Force Message = ENABLE). If you set Force Message to DISABLE, the message is written to the USER screen, but the screen does not change to the USER screen automatically. If you set Force Message to ENBL (TP OFF), then the user page is displayed automatically only if the teach pendant ENABLE (ON/OFF) keyswitch is set to OFF.	

4.2 SYSTEM CONFIGURATION SETUP PROCEDURE

Use Procedure 4-1 to set up items on the System Configuration Setup screen.

Procedure 4-1 System Setting

- | | |
|-------------|---|
| Step | <ol style="list-style-type: none"> 1 Press MENUS. 2 Select SYSTEM. 3 Press F1, [TYPE]. 4 Select Config. You will see a screen similar to the following. |
|-------------|---|

```

System/Config                                JOINT 10%
                                           4/27
1: Use HOT START:                            TRUE
2: I/O power fail recovery: RECOVER ALL
3: Autoexec program                          [*****]
   for Cold start:
4: Autoexec program                          [*****]
   for Hot start:
5: HOT START done signal:                    DO[ 0 ]
6: Restore selected program                  TRUE
7: Enable UI signals:                       FALSE
8: START for CONTINUE only:                 FALSE
9: CSTOPI for ABORT:                        FALSE
10: Abort all programs by CSTOPI:           FALSE
11: PROD_START depend on PNSTROBE:         FALSE
12: Detect FAULT_RESET signal               FALL
13: Use PPABN signal:                       <*GROUPS*>
14: WAIT timeout:                           30.00 sec
15: RECEIVE timeout                         30.00 sec
16: Return to top of program:                TRUE
17: Original program name(F1): [*****]
18: Original program name(F2): [*****]
19: Original program name(F3): [*****]
20: Original program name(F4): [*****]
21: Original program name(F5): [*****]
22: Default logical command: <*DETAIL*>
23: Maximum of ACC instruction:              500
24: Minimum of ACC instruction:              0
25: WJNT for default motion:                ADD
26: Auto display of alarm menu:             FALSE
27: Force message:                          ENABLE

[ TYPE ]

```


- 5 Move the cursor to the item you want to set, then enter a value by using the numeric keys or function keys on the teach pendant. Refer to Table 4-1 for detailed information about each field on the screen.
- To type a character string, move the cursor to the item, then press the ENTER key. You can then enter the necessary characters.
 - To set abnormal-pressure detection or default logic instructions, move the cursor to <*GROUPS*> or <*DETAIL*> then press ENTER. The screen for setting the corresponding item appears. Pressing the PREV key on that screen displays the System Config screen.
- 6 If you set or modify an item that requires a cold start to take effect, the following message appears on the screen “please power on again”. You must perform a cold start. Turn off the controller and then turn it on again. See the following screen for an example.

System/Config	JOINT 10%
	12/24
12: Detect FAULT_RESET signal	FALL
13: Use PPABN signal:	<*GROUPS*>
14: WAIT timeout:	30.00 sec
Please power on again	
[TYPE]	

**5 PLANNING AND
CREATING A PROGRAM**

5

PLANNING AND CREATING A PROGRAM

Topics In This Chapter		Page
Planning a Program	Before you write a program, you should plan the program. Planning involves considering the best way possible to perform a specific task before programming the robot to complete that task. Planning before creating a program will help you choose the appropriate instructions to use when writing the program.	5-2
	• Motion	5-2
	• Predefined Positions	5-4
Writing and Modifying a Program	Write a program using a series of menus on the teach pendant that allow you to add each instruction to your program. If the program sequence requires you to define the current location of the robot you jog or move the robot to the desired location and execute the appropriate instruction.	5-9
	• Writing a New Program	5-10
	• Modifying a Program	5-17
Modifying a Program in the Background (Background Editing)	After you create a program, you can modify the program. You can use a series of teach pendant screens to change or remove an instruction, add a new instruction, move instructions from one location in the program to another, or find specific sections of the program.	
	If the instruction requires defining the current location of the robot you jog or move the robot to the desired location and add the appropriate instruction. . . .	5-30
	• Background Edit Process Flow	5-31
	• Troubleshooting Background Edit	5-35

A FANUC Robotics *HandlingTool* program includes a series of commands, called instructions, that tell the robot and other equipment how to move and what to do to perform the palletizing task. For example, a program directs the robot and controller to:

- Move the robot in an appropriate way to required locations in the workcell.
- Perform an operation, such as pick up or put down a case.
- Send output signals to other equipment in the workcell.
- Recognize and respond to input signals from other equipment in the workcell.
- Keep track of time, part count, or job number.

5.1 PLANNING A PROGRAM

This section provides hints that will help you program specific tasks more efficiently. It contains hints for programming

- Motion
- Predefined positions
- Palletizing

NOTE The hints in this section are for programming, not jogging. World frame or user frame are usually best for jogging and recording positions.

5.1.1 Motion

Use the guidelines in this section to help you program certain kinds of robot motion.

Use Joint for the Fastest Move

Use joint motion type for the fastest moves and the shortest cycle time. Linear motion type results in slower moves. Use linear motion type only when it is the only way to move to a certain position.

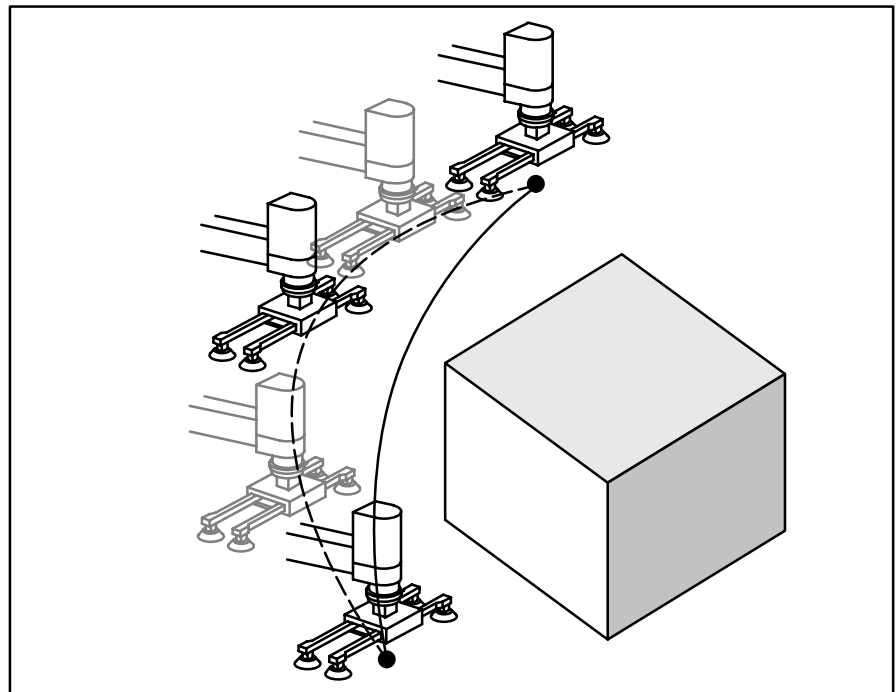
Use Fine at the End of a Task

Use fine termination type at the end of a task. Fine stops the robot precisely at the programmed point.

Use Continuous to Move Around Obstructions

Use continuous termination type for the most efficient movement around obstacles. Program the position near the obstacle and then adjust the termination type value as needed. See Figure 5-1.

Figure 5-1. Continuous Termination Type for Palletizing



Minimize Changes in Wrist Orientation

Abrupt changes in wrist orientation can sometimes increase cycle time. Smooth, gradual changes are faster and more efficient. Plan motions to minimize wrist orientation changes when possible.

When changes are necessary, distribute the rotation of the wrist smoothly throughout several moves; do not force the robot to change the entire wrist orientation at the end point of a series of moves.

To achieve maximum smoothness when changing wrist orientation,

1. Move the robot to the first position. Be sure that the orientation of the wrist is correct for that position.
2. Add a motion instruction to record the position of the robot.
3. Jog the robot to the last position that will be recorded. Be sure the robot is in the correct orientation for this position.
4. Add a motion instruction to record the position of the robot.
5. Record the number of motion instructions you think are necessary to perform the application between the first position and the last position.
6. At a slow speed, step forward toward the P[1] position. Refer to Section 7.2.2 for more information on single step motion.
7. At a slow speed, step forward toward the P[2] position.
8. **Before the robot reaches P[2]** to release the SHIFT or FWD key, press HOLD. Record this new position between P[1] and P[2].
9. Repeat Steps 7 and 8 for all the positions between the first position and the last position.

 CAUTION

If, while smoothing the motion, axes 4, 5, and 6 attempt to reach zero degrees and axis 5 flips orientation, change the jog coordinate system to JOINT and continue jogging.

5.1.2 Predefined Positions

You can use predefined positions in a program. A predefined position is a position you define that can be used several times in a program or in other programs.

Robot motion to a predefined position is often tied to an input signal. The robot must wait until the input signal is received before it can move to the predefined position. This allows the system to control when the robot is allowed to move to the predefined position.

An output signal can also be used to indicate to the system that the robot has reached the predefined position.

There are three ways to define a predefined position:

- Macros – These use position registers that contain the predefined position.
- Programs – These also use position registers that contain the predefined position.
- Position registers
- Reference Positions – Refer to Chapter 10 for information about reference positions.

Macros

Macros are programs that can be executed from:

- Specific teach pendant keys
- The MACROS screen
- The MANUAL FCTNS macro screen
- Within a program using the MACRO COMMAND instruction

Macros allow you flexibility as to how and when the robot moves to the predefined position. For example, you could specify a macro from the MACROS screen to move the robot to the home position automatically. Refer to Section 3.10.1 for information about setting up macros. Refer to Chapter 3 for information about setting up macros. Refer to Chapter 7 for information about executing macros.

Programs

You can write a program that moves the robot to a predefined position. You can use the CALL program instruction to branch to the program that moves the robot to the predefined position. Refer to Section 6.10.2 for more information.

Position Registers

Position registers can be used as predefined positions. Each position register can only contain one robot position. Refer to Section 6.8 for more information about position registers.

To set up a position register as a predefined position,

1. Press DATA.
2. Press F1, [TYPE].
3. Select Position Reg.
4. Jog the robot to the position you want to predefine as your perch position.
5. Hold down the SHIFT key and press F3, RECORD.

To use a position register as a predefined position in a program, include the position register you set up in a motion instruction. For example,

L PR[1] 50mm/sec FINE

Predefined positions that are commonly used are:

- Home or perch position
- Repair position
- Safe position (At Pounce)

CAUTION

Recorded positions are not affected by UFRAME and UFRAME has no effect during playback. However, **position registers are recorded with respect to UFRAME**. If you change UFRAME, any recorded position registers will also change.

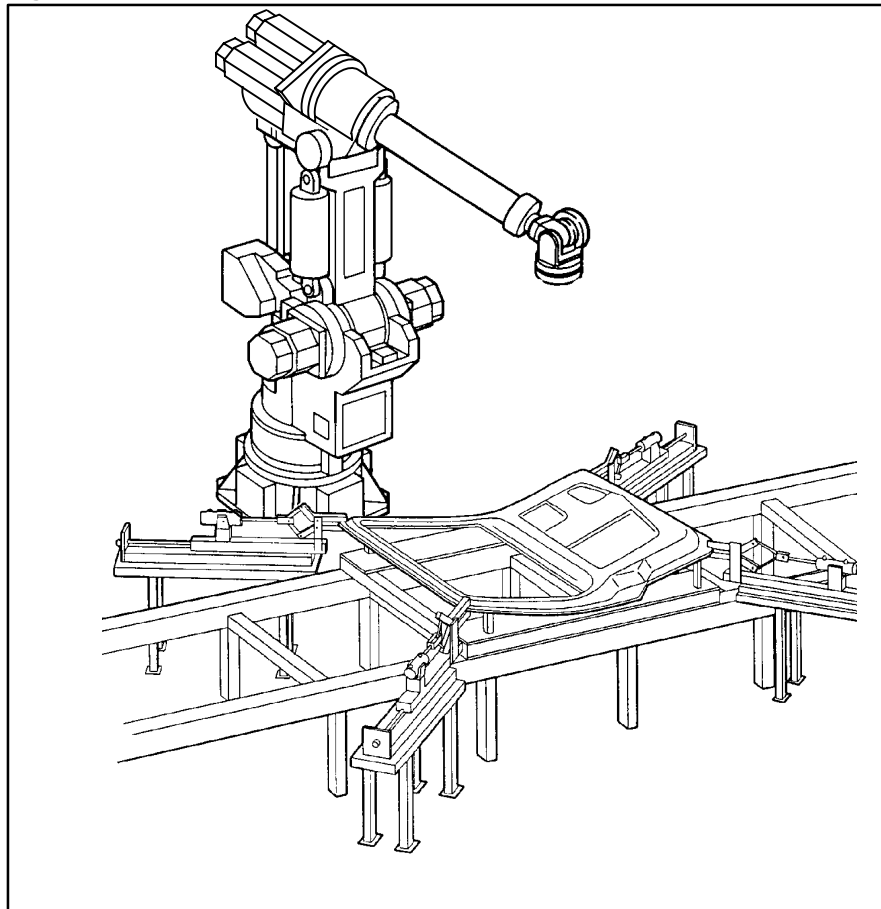
NOTE If the position register is to be shared between two programs, both programs should have the same tool and UFRAME, in order to move the robot to the same position in space.

Home Position (Perch)

The home position (Perch), is a position away from the workpiece transfer area. Program the robot to move to home before the first position, between cycles, and any time the robot must be away from workcell activity.

Figure 5-2 shows an example of a home position.

Figure 5-2. Home Position



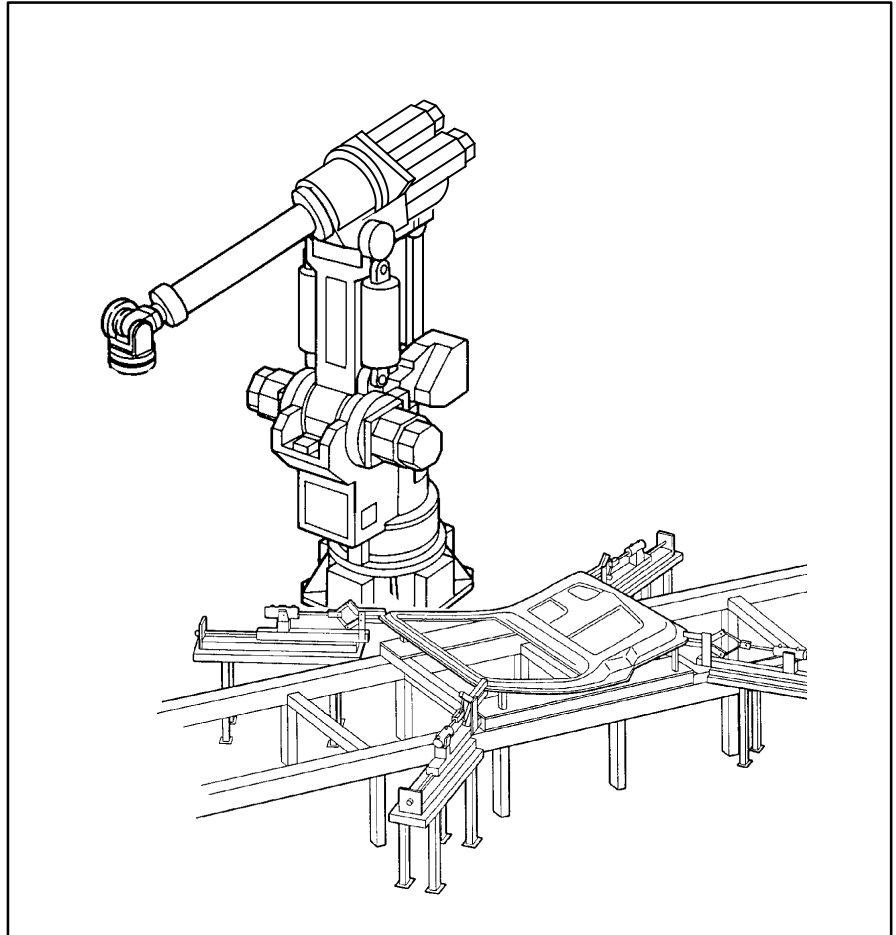
NOTE “Home Position” is an input signal in UOP not a predefined position. It can be one of the Reference Positions, but these are not utilities or built-ins used to send the robot to the “HOME” position.

Repair Position

The repair position is a position where robot repair operations are performed. Program the robot to move to the repair position any time repair operations must be performed. Record the repair position away from other equipment and the transfer area.

Figure 5-3 shows an example of a repair position.

Figure 5-3. Repair Position



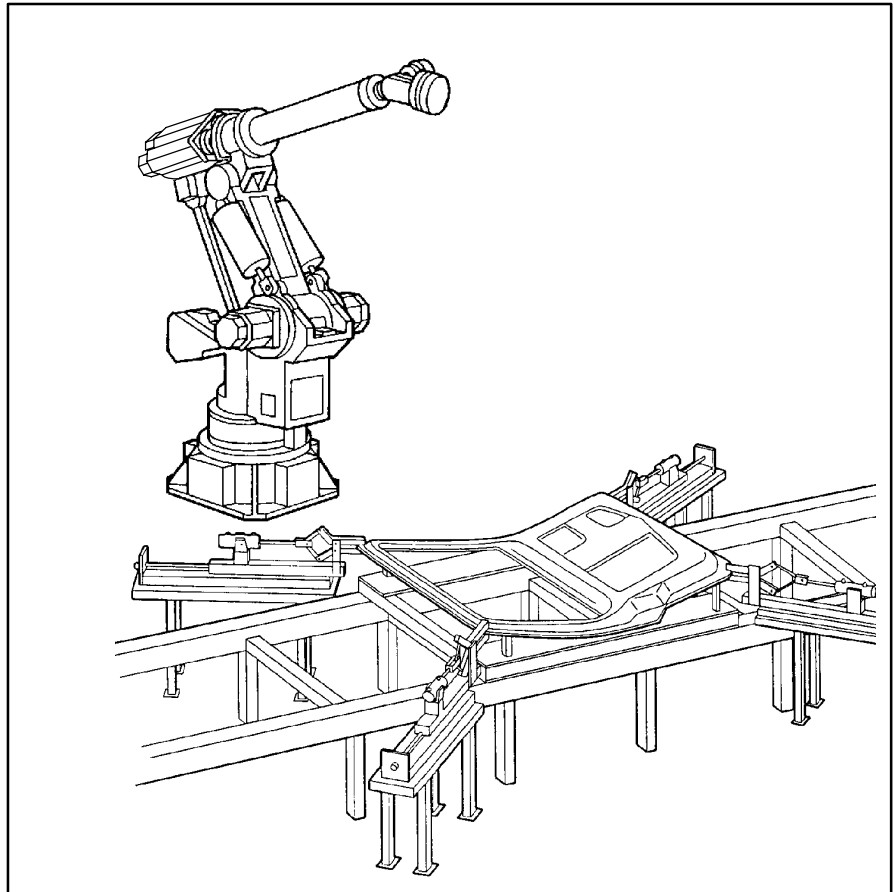
NOTE “Repair Position” is not a predefined position, but you can use one of the Reference Positions for this purpose. A user program with the same taught position must be run to move the robot to the “REPAIR” position.

Safe Position (Pounce)

The safe position, (Pounce), is away from fixtures and the workpiece transfer area. Program the robot to move to the safe position any time it is necessary to move the robot away from other workcell activities.

Figure 5-4 shows an example of a safe position.

Figure 5-4. Safe Position



NOTE “Safe Position” is not a predefined position, but you can use one of the Reference Positions for this purpose. A user program with the same taught position must be run to move the robot to the “SAFE” position.

Other Positions

You can define any other positions to be predefined positions. Define any position that the program uses more than once as a predefined position. This minimizes the time it takes to create and modify your program.

5.2 WRITING AND MODIFYING A PROGRAM

You can write new programs and modify existing programs to direct the robot to perform a task. Writing a program includes:

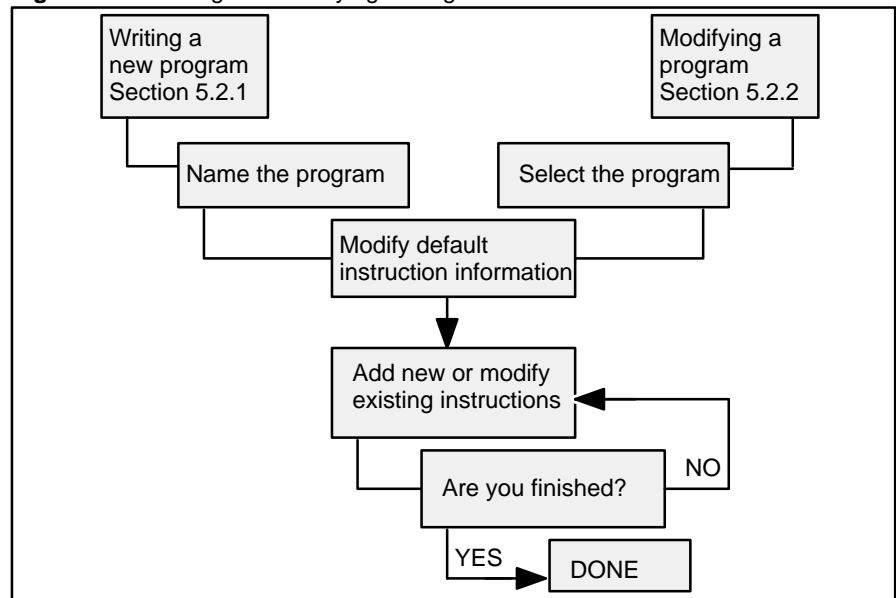
- Naming the program
- Defining default instructions
- Adding instructions to the program

Modifying a program includes:

- Selecting the program
- Modifying default instructions
- Inserting instructions
- Deleting instructions
- Copying and pasting instructions
- Searching for instructions
- Renumbering instructions
- Undoing operations
- Displaying comments

Figure 5-5 summarizes writing and modifying a program.

Figure 5-5. Writing and Modifying a Program



5.2.1 Writing a New Program

When you write a new program you must

- **Name** the program.
- **Modify** default instruction information. This includes modifying motion instructions and other instructions.
- **Add** motion instructions to the program.
- **Add** other instructions to the program.

Use Procedure 5-1 to create and write a new program.

Naming the Program

You can name a program using three different methods (refer to Section 6.1.5 for additional information):

- **Words** – Selected predefined words to insert in the menu. the predefined words are:
 - RSR
 - PNS
 - MAIN
 - SUB
 - TEST
- **Upper Case** – This method lets you use upper case letters and any numbers.
- **Lower Case** – This method lets you use lower case letters and any numbers. For the program name, lower case letters are automatically converted to upper case after you enter them.

Options allows you to change whether you are overwriting, inserting, or clearing the program name or comment information. The screen will display either Insert or Overwrite. Clear allows you to remove text from the current field.

The total length of the program name must be no more than eight characters. You can combine words, upper case letters, and lower case letters to form the program name. Give the program a unique name that indicates the purpose of the program.

NOTE Do not use the asterisk * symbol in program names.

Defining Detail Information

The detail of program header information includes:

- Creation date
- Modification date
- Copy Source
- Number of positions and program size
- Program Name
- Sub Type
- Comment
- Group mask
- Write protection
- Ignore pause

Refer to Section 6.1 for details about program header information.

Defining Default Instruction Information

Motion instructions tell the robot to move to an area in the workcell in a specific way. When you create a program you can define, in advance, the way you want the robot to move when you add a motion instruction. You do this by defining default motion instruction information. Default motion instructions can include arc welding instructions.

After you have defined the default instructions you can add them to the program. You select one of the available default instructions to be the current default instruction by moving the cursor to that instruction. You can define and change default instructions any time while writing or modifying a program.

Modifying Default Motion Statements by Adding Option Fields (adding instructions)

You can also add other instructions not included in the default motion instruction to your program. To add these instructions, select the kind of instruction you want to add to the program and use the information on the screen to enter specific instruction information.

You add all instructions using the same general procedure. Motion instructions, however, require some specific information. Refer to Procedure 5-1 for information on adding motion and other kinds of instructions.

Procedure 5-1 Creating and Writing a New Program

- Condition**
- All personnel and unnecessary equipment are out of the workcell.
 - The teach pendant is turned on.

Naming the Program

- 1 Press SELECT.
- 2 If F2, CREATE, is not displayed, press NEXT, >.
- 3 Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
- 4 Press F2, CREATE. You will see a screen similar to the following.

```

                                JOINT 10%
1 Words
2 Upper Case
3 Lower Case
4 Options          -- Insert --
Select
--- Create Teach Pendant Program ---
Program Name [█      ]
                                -- End --
Enter program name
RSR      PNS      MAIN      SUB      TEST

```

Program Name [█]

- 5 Enter the program name:

NOTE If you are writing a program for production operation using RSR or PNS, name the program as follows:

- An **RSR** program must be **RSRnnnn** where **nnnn** is a four-digit number, such as; RSR0001.
- A **PNS** program must be **PNSnnnn**, where **nnnn** is a four-digit number, such as; PNS0001.
 - a Move the cursor to select a method of naming the program: Words, Upper Case, or Lower Case.
 - b Press the function keys whose labels correspond to the name you want to give to the program. These labels vary depending on the naming method you chose in Step a.

For example, if you chose Upper Case, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the program name field. Press the right arrow key to move the cursor to the next space. Continue until the entire program name is displayed.

To delete a character, press BACK SPACE.

- c When you are finished, press ENTER. You will see a screen similar to the following.

```

                                JOINT 10%
1 Words
2 Upper Case
3 Lower Case
4 Options
Select

--- Create Teach Pendant Program ---

Program Name [RSR1000      ]

                                -- End --

Select function
                DETAIL      EDIT

```

- 6 To display program header information, press F2, DETAIL. You will see a screen similar to the following.

```

Program Detail                                JOINT 10%
                                                    1/6
Creation Date:                                02-Jan-9x
Modification Date:                            02-Jan-9x
Copy Source:      [                               ]
Positions:  10   Size:      312 Byte
1 Program Name:      [ RSR1000 ]
2 Sub Type:          [NONE      ]
3 Comment:           [                               ]
4 Group Mask:        [1,*,*,*,*]
5 Write protect:     [ON         ]

END      PREV      NEXT

```

To skip setting program header information and begin editing the program, press F1, END, and skip to **Defining Default Motion Instructions** in this procedure.

- 7 To set or rename the program, move the cursor to the program name and press ENTER.
- Move the cursor to select a method of naming the program: Words, Upper Case, or Lower Case.
 - Press the function keys whose labels correspond to the name you want to give to the program. These labels vary depending on the naming method you chose in Step a. **To delete a character**, press BACK SPACE.
 - When you are finished, press ENTER.

- 8 To select a sub type**, move the cursor to the sub type and press F4, [CHOICE]. You will see a screen similar to the following. Refer to Section 6.1.6 for more information on sub types.

Sub Type		JOINT 10%
1	None	
2	Macro	
Program Detail		JOINT 10%
1	Program Name	[PROC742]
2	Sub Type:	[]
3	Comment:	[]
4	Group mask:	[1, *, *, *, *]
5	Write protect:	[OFF]
[CHOICE]		

- a Select whether the sub type is None or Macro.
 - b Press ENTER.
- 9 To type a comment**, move the cursor to Comment and press ENTER.
- a Select a method of naming the comment.
 - b Press the appropriate function keys to add the comment.
 - c When you are finished, press ENTER.

For example, if you chose Upper Case, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire comment is displayed.

- 10 To set the group mask (or motion group)**, move the cursor to the group you want to enable or disable. You can use multiple groups in a single program, but only two groups can perform Cartesian motion within a single program. The first position in the group mask corresponds to the first group. Only groups 1, 2, and 3 are currently available. If the program does not contain any motion statements, use *. This will allow another program to run which locks the group.
- a **To enable a group**, scroll right to enable Group 2, 3, and so forth. Press F4, 1 for each enabled group.
 - b **To disable a group**, press F5, *. If you disable all groups, you cannot add motion instructions to your program.

NOTE If your system is not set up for multiple groups, you will only be able to select a 1, for the first group, or a *, for no group.

NOTE After the group mask has been set, and motion instructions have been added to the program, the group mask cannot be changed for that program.

- 11 To set write protection**, move the cursor to Write protect. Refer to Section 6.1.9 for information on write protect.
 - a** To turn write protection on, press F4, ON.
 - b** To turn write protection off, press F4, OFF.
- 12 To set ignore pause**, move the cursor to Ignore pause. Refer to Chapter 6 for information on ignore pause.
 - a** To turn on ignore pause, press F4, ON.
 - b** To turn off ignore pause, press F4, OFF.

NOTE Whenever you want to return to the first SELECT menu, press PREV until it is displayed.

- 13** Turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.
- 14 When you have finished entering program information**, press F1, END. The teach pendant editor screen will be displayed.

Background Program Editing

Defining Default Motion Instructions

To edit a program in the *background*, with the teach pendant off, select the program called “-BCKEDT-”. You will be asked to select a program to edit in the background. **Refer to Section 5.3 for more information.**

- 1** Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
- 2** Press EDIT.
- 3** Press F1, POINT.

You will see a list of default motion instructions.

- Two for joint motions
- Two for linear motions

NOTE If the instructions listed are the ones you want to use, do not modify them.

- 4** To modify the default motion instructions, press F1, ED_DEF.
- 5** Move the cursor to the default instruction you want to modify.
- 6** Move the cursor to the component you want to modify.
- 7** Use the appropriate arrow and function keys to modify the component and press ENTER.

If the CHOICE function key is displayed, press F4 to display a list of values for the selected component.

Joint default menu

1: J P[] 100% FINE

2: J P[] 100% CNT100

3: L P[] 100 mm/sec FINE

4: L P[] 100 mm/sec CNT100

```
1 J P[ ] 100% CNT50
1 J P[ ] 50% CNT50
```

```
DONE
F5
```

```
[ INST ] [ EDCMD ]
F1
```

When You Are Finished

For example, to change the speed value, move the cursor to 100%. Type a new value and press ENTER. The new value will be displayed. Each time you add this instruction to the program the new value will be used.

- 8 Repeat Steps 5 through 7 for each default instruction that you want to define.
- 9 **When you are finished defining default motion instructions**, move the cursor to the instruction you want to be the current default instruction and press F5, DONE.
- 10 **To save the modified default motion instructions**, refer to Section 9.3.3, “Backing Up Program System and Application Files.”
- 11 **To load default motion instruction files**, refer to Section 9.3.2, “Loading Files From Disk to Controller Memory.”
- 12 **To add other instructions**, press NEXT until F1, [INST] is displayed then press F1, [INST]. Select the kind of instruction you want and use the appropriate selections on the screen to build the instruction.

Refer to Chapter 6 for details about each instruction.
- 1 Turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.

5.2.2 Modifying a Program

You can modify an existing program any time you want to change the content of the program. Modifying a program includes

- **Selecting** a program
- **Modifying** motion instructions
- **Modifying** palletizing instructions
- **Modifying** other instructions
- **Inserting** instructions
- **Deleting** instructions
- **Copying and pasting** an existing instruction or program element
- **Finding and replacing** an existing instruction or program element
- **Renumbering** positions after instructions have been added, removed, or moved
- **Undoing** operations, such as modifying instructions, inserting instructions, and deleting instructions
- **Displaying** comments on the teach pendant program screen

Selecting a Program

Selecting a program allows you to choose the program name from a list of existing programs in controller memory. Refer to Chapter 9 for more information on loading programs.

Touching Up and Modifying Motion Instructions

Touching up motion instructions changes any element of the motion instruction. The element you might modify most often is the position data.

Modifying Other Instructions

Modifying other instructions changes any element of the instruction.

Inserting Instructions

Inserting instructions places a specified number of new instructions between existing instructions. When you insert an instruction, the instructions that follow the new instruction are renumbered automatically.

Deleting Instructions

Deleting instructions removes them from the program permanently. When you remove an instruction the remaining instructions are renumbered automatically.

Copying and Pasting Instructions

Copying and pasting allows you to select a group of instructions, make a copy of the group, and insert the group at one or more locations in the program. You can paste copied program instructions using the methods described in Table 5-1.

Table 5-1. Paste Methods

Copied program instructions:		
1: J P[1] 100% CNT100 2: L P[2] 500mm/sec CNT80 3: DO[1] = ON 4: J P[3] 50% CNT50 5: L P[4] 10mm/sec FINE 6: MOVE HOME		
Method	Description	Pasted Program Instructions
LOGIC (F2)	<ul style="list-style-type: none"> Pastes the lines exactly as they were copied Does not record positions Leaves the position numbers blank 	7: J P[...] 100% CNT100 8: L P[...] 500mm/sec CNT80 9: DO[1] = ON 10: J P[...] 50% CNT50 11: L P[...] 10mm/sec FINE 12: MOVE HOME
POS_ID (F3)	<ul style="list-style-type: none"> Pastes the lines exactly as they were copied Retains the original position numbers 	7: J P[1] 100% CNT100 8: L P[2] 500mm/sec CNT80 9: DO[1] = ON 10: J P[3] 50% CNT50 11: L P[4] 10mm/sec FINE 12: MOVE HOME
POSITION (F4)	<ul style="list-style-type: none"> Pastes the lines exactly as they were copied Renumbers the copied positions with next available position numbers Retains copied positional data 	7: J P[5] 100% CNT100 8: L P[6] 500mm/sec CNT80 9: DO[1] = ON 10: J P[7] 50% CNT50 11: L P[8] 10mm/sec FINE 12: MOVE HOME
CANCEL (F5)	Cancels the paste and retains the copied lines so you can paste them elsewhere	
R-LOGIC (NEXT+F2)	<ul style="list-style-type: none"> Pastes the lines in reverse order Does not record positions Leaves the position numbers blank 	7: MOVE HOME 8: L P[...] 10mm/sec FINE 9: J P[...] 50% CNT50 10: DO[1] = ON 11: L P[...] 500mm/sec CNT80 12: J P[...] 100% CNT100
R-POS-ID (NEXT+F3)	<ul style="list-style-type: none"> Pastes the lines in reverse order Retains the original position numbers 	7: MOVE HOME 8: L P[4] 10mm/sec FINE 9: J P[3] 50% CNT50 10: DO[1] = ON 11: L P[2] 500mm/sec CNT80 12: J P[1] 100% CNT100
R-POS (NEXT+F4)	<ul style="list-style-type: none"> Pastes the instructions in reverse order Renumbers the copied positions with the next available position numbers 	7: MOVE HOME 8: L P[8] 10mm/sec FINE 9: J P[7] 50% CNT50 10: DO[1] = ON 11: L P[6] 500mm/sec CNT80 12: J P[5] 100% CNT100

Table 5-1. (Cont'd) Paste Methods

Method	Description	Pasted Program Instructions
RM-POS-ID (NEXT+F3)	<ul style="list-style-type: none"> ● Pastes all instructions except motion instructions in reverse order ● Motion instructions are created using the current (first) and next (second) motion instruction: <ul style="list-style-type: none"> – Motion type, speed, and motion-related options are used from the second motion instruction – Positional data, termination type, and non-motion-related options are used from the first motion instruction – The bottom motion instruction in the copied area is pasted as is ● Retains the original position numbers ● Modal instructions, such as I/O, are pasted in reverse order, but the states are not changed (ON to OFF, or OFF to ON); you must modify these after the paste ● UTOOL and UFRAME change instructions and position register instructions are pasted in reverse order, but the frame numbers and register numbers are not changed; you must modify these after the paste ● Motion instructions that contain the following motion options are only copied in reverse order and cause a warning to be displayed: <ul style="list-style-type: none"> – Application commands (such as SPOT) – SKIP – INC (incremental motion) – Continuous turn – TIME BEFORE/AFTER 	<pre> 1st: J P[i] 100% CNT100, Offset,PR[x] 2nd: L P[j] 50 mm/sec FINE ACC150 PTH Result: L P[j] 50mm/sec FINE ACC150 PTH L P[i] 50mm/sec CNT100 Offset,PR[x] ACC150 PTH 7: MOVE HOME 8: L P[4] 10mm/sec FINE 9: L P[3] 10mm/sec CNT50 10: DO[1] = ON 11: J P[2] 50% CNT80 12: L P[1] 500mm/sec CNT100 </pre>
RM-POS (NEXT+F5)	<ul style="list-style-type: none"> ● Pastes all instructions except motion instructions in reverse order ● Motion instructions are created using the current (first) and next (second) motion instruction: <ul style="list-style-type: none"> – Motion type, speed, and motion-related options are used from the second motion instruction – Positional data, termination type, and non-motion-related options are used from the first motion instruction – The bottom motion instruction in the copied area is pasted as is ● Renumbers the copied positions with the next available position numbers ● Modal instructions, such as I/O, are pasted in reverse order, but the states are not changed (ON to OFF, or OFF to ON); you must modify these after the paste ● UTOOL and UFRAME change instructions and position register instructions are pasted in reverse order, but the frame numbers and register numbers are not changed; you must modify these after the paste ● Motion instructions that contain the following motion options are only copied in reverse order and cause a warning to be displayed: <ul style="list-style-type: none"> – Application commands (such as SPOT) – SKIP – INC (incremental motion) – Continuous turn – TIME BEFORE/AFTER 	<pre> 1st: J P[k] 100% CNT100, Offset,PR[x] 2nd: L P[l] 50 mm/sec FINE ACC150 PTH Result: L P[l] 50mm/sec FINE ACC150 PTH L P[k] 50mm/sec CNT100 Offset,PR[x] ACC150 PTH 7: MOVE HOME 8: L P[8] 10mm/sec FINE 9: L P[7] 10mm/sec CNT50 10: DO[1] = ON 11: J P[6] 50% CNT80 12: L P[5] 500mm/sec CNT100 </pre>

Finding and Replacing Instructions

Finding and replacing is finding specific instructions and, if desired, replacing those instructions with new instructions. This function is useful, for example, when setup information that affects the program is changed. It is also useful when you need to find a specific area of a long program quickly.

Renumbering Positions

Renumbering allows you to renumber positions in the program. When you add positions to a program, the position number is incremented from the last numbered position regardless of its place in the program. When you delete motion instructions, all remaining positions keep their current numbers. Renumbering reassigns all position numbers in the program so that they are in sequential order.

Undoing Operations

Undoing operations allows you to recover from the last incorrect or faulty operation. You can undo the previous operation to edit the program, and you can redo the last Undo operation. Several modifications for one line can be recovered with a single Undo operation if you have not modified any other line in between modifications.

You cannot Undo an operation when any of the following conditions exist:

- The controller has been turned off and turned on
- Another program has been selected
- You are using a CRT and keyboard
- The teach pendant ON/OFF switch is OFF
- The program is write-protected
- The teach pendant does not have enough available memory

In addition, Undo will not work if you have executed any of the following instructions in a program:

- Line tracking instructions
- On-line touch up

NOTE If power fails in the process of performing an Undo operation, unexpected results can occur, and the desired modification is not guaranteed to have taken effect when power is restored.

Displaying Comments on the Teach Pendant Program Screen

This function displays comments of I/O and registers on the teach pendant program screen. This gives you the ability to confirm comments while you edit a teach pendant program, without having to display another screen.

You can only **display** comments for the following instructions while you are editing a teach pendant program. You cannot change the comments on the teach pendant program screen.

- I/O instructions
(DI[i:COMMENT], DO[i:COMMENT], RI[i:COMMENT], RO[i:COMMENT], GI[i:COMMENT], GO[i:COMMENT], AI[i:COMMENT], AO[i:COMMENT], UI[i:COMMENT], UO[i:COMMENT], SI[i:COMMENT], SO[i:COMMENT])
- Register instructions (R[i:COMMENT])
- Position register instructions (PR[i:COMMENT])
- Pallet register instructions (PL[i:COMMENT])
- Position register element instructions (PR[i.j:COMMENT])

The display of the comments is turned on and off when you select the [Comment] by pressing F5, [EDCMD].

NOTE Instructions that include indirect addressing do not display the comments, as in the following example:

```
R[R[i]], DO[R[i]], ...
```

Use Procedure 5-2 to modify a program.

Procedure 5-2 Modifying a Program

- Condition**
- All personnel and unnecessary equipment are out of the workcell.
 - The program has been created and all detail information has been set correctly. (Procedure 5-1)

Selecting a Program

- 1 Press SELECT.
- 2 Display the appropriate list of programs:
 - a Press F1, [TYPE].
 - b Select the list you want:
 - **All** displays all programs.
 - **TP Programs** displays all teach pendant programs.
 - **Macro** displays all macro programs.
- 3 Move the cursor to the name of the program you want to modify.
- 4 Press EDIT.
- 5 Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

CAUTION

Recorded positions are not affected by UFRAME and UFRAME has no effect during playback. However, **position registers are recorded with respect to UFRAME**. If you change UFRAME, any recorded position registers will also change.

Touching Up and Modifying Motion Instructions

- 1 Move the cursor to the **line number** of the motion instruction you want to modify.
- 2 **To change only the position component of the motion instruction**, jog the robot to the new position, press and hold in the SHIFT key and press F5, TOUCHUP.

Position has been recorded to
P[n]

3 To change other motion instruction components, move the cursor to the component using the arrow keys, and press the appropriate function keys to modify the component:

- **If function key labels are available**, press the appropriate one.
- **If no function key labels are available**, press F4, [CHOICE], and select a value.
- **To change the position value**, move the cursor to the position number and press F5, POSITION. The position screen will be displayed showing the Cartesian coordinates or joint angles of the selected position. Move the cursor to the component you want to change and enter the new value using the number keys. To make other changes, use the function keys, described here.

1: J P[6] 100% CNT50

Position Detail		JOINT 30%	
P[1]	UF:0 UT:1	CONF: N	0 0
X	1829.992 mm	W	-179.998 deg
Y	.050 mm	P	-90.000 deg
Z	1170.024 mm	R	0.000 deg
PRG1		2/4	

GROUP PAGE CONFIG DONE [REPRE]
F1 F2 F3 F4 F5

PAGE POSITION DONE [REPRE]
F1 F2 F3 F4 F5

- **To change the motion group number**, press F1, GROUP. This applies only to systems that have been set up for multiple groups.
- **To display components for extended axes**, press F2, PAGE. This only applies to systems that include extended axes.
- **To change the configuration** between flip (F) and no-flip or normal (N), press F3, CONFIG, and then use the up and down arrow keys to change F to N and N to F.

NOTE Joint angles are useful for zero-positioning the robot or for non-kinematic motion control such as controlling the motion of a positioning table.

- **To change the format of the position from Cartesian coordinates to joint angles or from joint angles to Cartesian coordinates**, press F5, [REPRE] and select the coordinate system. The position is converted automatically.
- When you are finished, press F4, DONE.

CAUTION

When you change the representation from Cartesian to joint, the user frame and tool frame values currently in effect will be used to calculate the joint angles. After you have changed from Cartesian to joint, if you modify user frames and tool frames, these changes will have no effect on the position with joint representation, regardless of the motion type of the motion instruction that contains the position.

4 Repeat Steps 1 through 3 for each motion instruction you want to modify.

Modifying Other

```
2 P[3]=15
```

- 1 Move the cursor to the line number of the instruction you want to modify.
- 2 Move the cursor to the component you want to modify and press the appropriate key:
 - **If function key labels are available**, press the appropriate one.
 - **If no function key labels are available**, press F4, [CHOICE], and select a value.
- 3 Repeat Steps 1 and 2 for each instruction you want to modify.

Inserting Instructions

```
5: P[4] 50% CNT50
6 J P[5] 50% CNT50
```

```
[INST] [EDCMD]
```

```
F5
```

```
1 Insert
```

```
How many lines to insert?:
```

```
5: J P[4] 50% CNT50
6:
7:
8:
9: J P[5] 50% CNT10
```

- 1 Decide where you want to insert the instruction. Move the cursor to the line following that point. The cursor must be on the **line number**. For example, if you want to insert between lines 5 and 6 place the cursor on line 6.
- 2 Press NEXT, >, until F5, [EDCMD] is displayed.
- 3 Press F5, [EDCMD].
- 4 Select 1, Insert.
- 5 Type the number of lines to insert and press ENTER.

A blank line will be inserted into the program for each line you want inserted. All lines in the program will be automatically renumbered.

- 6 Move the cursor to the line number of any inserted line and add any instruction.
- 1 Move the cursor to the line number of the instruction you want to delete. If you want to delete several instructions in consecutive order, move the cursor to the first line to be deleted.

Deleting Instructions

```
5 J P[4] 50% CNT50
6 J P[5] 50% CNT10
7 J P[6] 75% CNT10
```

CAUTION

Deleting an instruction permanently removes the instruction from the program. Be sure you want to remove an instruction before you continue; otherwise, you could lose valuable information.

```
[INST] [EDCMD]
```

```
F5
```

```
2 Delete
```

```
5 J P[4] 50% CNT50
6 J P[5] 50% CNT10
7 J P[6] 75% CNT10
```

```
Delete line(s) ?
```

```
YES NO
```

- 2 Press NEXT, >, until F5, [EDCMD], is displayed.
- 3 Press F5, [EDCMD].
- 4 Select 2, Delete.
- 5 **To delete a range of lines**, move the cursor to select the lines to be deleted. The line number of each line to be deleted will be highlighted as you move the cursor.
- 6 Delete the line or lines:
 - **If you do not want to delete the selected line(s)**, press F5, NO.
 - **To delete the selected line(s)** press F4, YES.

Copying and Pasting Instructions

```
[INST]                                [EDCMD]
                                     F5
```

```
3 Copy
```

```
5J P[4] 50% CNT50
6J P[5] 50% CNT10
7J P[6] 75% CNT10
```

```
Paste before this line?
```

```
LOGIC POS-ID POSITION CANCEL>
  F2   F3   F4   F5
```

```
R-LOGIC R-POS-ID R-POSITION CANCEL>
  F2   F3   F4   F5
```

- 1 Press NEXT, > until F5, [EDCMD] is displayed.
- 2 Press F5, [EDCMD].
- 3 Select 3, Copy.
- 4 Move the cursor to the first line to be copied.
- 5 Press F2, COPY.
- 6 Move the cursor to select the range of lines to be copied.
The line number of each line to be copied will be highlighted as you move the cursor.
- 7 Press F2, COPY, again.
- 8 Decide where you want to paste the lines. Move the cursor to the line following that point. The cursor must be on the **line number**.
- 9 Press F5, PASTE.
- 10 Press the function key that corresponds to the way you want to paste the copied lines:
 - **LOGIC (F2)** – adds the lines exactly as they were, does not record positions, and leaves the position numbers blank.
 - **POS_ID (F3)** – adds the lines exactly as they were and retains the current position numbers.
 - **POSITION (F4)** – adds the lines exactly as they were and renumbers the copied positions with the next available position numbers. All positional data is transferred.
 - **CANCEL (F5)** – cancels the paste, but the copied lines are retained so you can paste them elsewhere.
 - **R-LOGIC (NEXT+F2)** – adds the lines in reverse order, does not record the positions, and leaves the position numbers blank.
 - **R-POS-ID (NEXT+F3)** – adds the lines in reverse order and retains their original position numbers.

- **RM-POS-ID (NEXT+F3)** – pastes all instructions except motion instructions in reverse order. Motion instructions are created using the current (first) and next (second) motion instruction: Original position numbers are retained. Refer to Table 5-1 for details and an example.
- **R-POS (NEXT+F4)** – adds lines in reverse order and renumbers the copied positions with the next available position numbers. Refer to Table 5-1 for details and an example.
- **RM-POS (NEXT-F5)** – pastes all instructions except motion instructions in reverse order. Motion instructions are created using the current (first) and next (second) motion instruction: The copied positions are renumbered with the next available position numbers. Refer to Table 5-1 for details and an example.

NOTE When you use RM-POS-ID and RM-POS, motion instructions that contain the following motion options are only copied in reverse order and cause a warning to be displayed:

- Application commands (such as SPOT)
- SKIP
- INC (incremental motion)
- Continuous turn
- TIME BEFORE/AFTER

11 Repeat Steps 6 through 10 to copy and paste the instructions.

12 **When you are finished copying and pasting instructions,** press PREV twice.

Finding Instructions

```
[EDCMD]
  F5
4 Find
```

```
  NEXT  EXIT
  F4    F5
```

Replacing Instructions

```
[INST] [EDCMD]
  F5
5 Replace
Select old item
```

```
Select new item
```

```
Replace OK?
  YES  NEXT  EXIT
  F3   F4   F5
```

- 1 Move the cursor to the line number of any instruction.
- 2 Press NEXT, >, until F5, [EDCMD], is displayed.
- 3 Press F5, [EDCMD].
- 4 Select 4, Find.
- 5 Select the type of instruction to find.
- 6 When prompted, enter the necessary information.

The system searches forward from the current cursor position for the item you want. If it finds an instance of the item, it highlights it on the screen.

- 7 **To find the next instance of the item**, press F4, NEXT.
- 8 **When you are finished finding items**, press F5, EXIT.
- 9 Press PREV twice.

NOTE You can not use the Find command to locate track/offset instructions or touch sense instructions.

- 1 Move the cursor to the line number of any instruction.
- 2 Press NEXT, > until F5, [EDCMD], is displayed.
- 3 Press F5, [EDCMD].
- 4 Select 5, Replace.
- 5 Select the instruction you want to replace from the list of instructions. Follow the information on the screen to specify the instruction.

The system finds the first instance of the existing instruction and highlights it.
- 6 Select the replacement item and enter the necessary information.
- 7 Decide how to replace the instruction:
 - **To replace the existing instruction with the new instruction** press F3, YES. The system will prompt you to search for the next one.
 - **To ignore this instance and find the next**, press F4, NEXT, and the system will find the next instance, if there is one.
 - **To stop the cancel and replace operation**, press F5, EXIT

CAUTION

You cannot use the *Replace* command to replace a motion instruction with a touch sense or track/offset instruction. Doing so causes a memory write failure error. If you want to replace the motion instruction, first delete the motion instruction and then insert the touch sense or track instruction instruction.

- 8 Press PREV twice.

Renumbering Positions

```
[ INST ]                [ EDCMD ]
                        F5
6 Renumber
Renumber OK ?
                        YES NO
```

Undoing Operations

```
[ INST ]                [ EDCMD ]
                        F5
7 Undo
Undo?(Insert)
                        YES NO
```

```
Redo?
                        YES NO
```

- 1 Move the cursor to the line number of any instruction.
- 2 Press NEXT, >, until F5, [EDCMD], is displayed.
- 3 Press F5, [EDCMD].
- 4 Select 6, Renumber.
- 5 Renumber the positions:
 - If you do not want to renumber positions press F5, NO.
 - To renumber positions press F4, YES.

- 1 Press NEXT, >, until F5, [EDCMD], is displayed.
- 2 Press F5, [EDCMD].
- 3 Select 7, Undo.
- 4 Undo the operation. The recovered operation is displayed.
 - If you do not want to undo the operation, press F5, NO.
 - To undo the operation, press F4, YES.
- 5 To cancel the undo operation, press F5, [EDCMD] , then press 7, Undo. Press YES to redo the operation.

⚠ WARNING

If you have used Undo, you should thoroughly test the modified program at a low motion speed before allowing it to run continuously. Otherwise, you could injure personnel or damage equipment. Refer to Section 7.2 for more information on testing a program.

Displaying Comments on the Teach Pendant Program Screen

- 1 See the following screen for an example.

```
PNS0001                JOINT 10 %
                        1/3
1: R[2:COUNTER1]=DI[3:HAND1]
2: DO[1:HAND1ACK]=ON
[END]
[ INST ]                [ EDCMD ]>
```

- 2 Press F5, [EDCMD].

```
PNS0001                JOINT 10 %
                        1/3
1: R[2:COUNTER1]=DI[3:HAND1]
2: DO[1:HAND1ACK]=ON
[END]
                        +-----+
                        | 1 Insert  |
                        | 2 Delete  |
                        | 3 Copy    |
                        | 4 Find    |
                        | 5 Replace  |
                        | 6 Renumber |
                        | 7 Comment  |
                        +-----+
[ INST ]                [ EDCMD ]
```

- 3** If you select Comment, the comments turn off.

```

PNS0001                                JOINT 10 %
                                         1/3
1:  R[2]=DI[3]
2:  DO[1]=ON
[END]
[ INST ]                                [ EDCMD ]>

```

- 4** Press F5, [EDCMD], again.
If you select Comment, the comments turn on.

```

PNS0001                                JOINT 10 %
                                         1/3
1:  R[2:COUNTER1]=DI[3:HAND1]
2:  DO[1:HANDLACK]=ON
[END]
[ INST ]                                [ EDCMD ]>

```

When You Are Finished

- 1** Turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.

5.3 MODIFYING A PROGRAM IN THE BACKGROUND (BACKGROUND EDITING)

Background editing is used to modify a program when the teach pendant is off. This can also be used to edit a program while another program is running. You do not have to stop the robot to modify or check another program. This option can improve the efficiency of production and maintenance.



WARNING

If the teach pendant is off, make sure you are at a safe distance (outside of the robot operating range) when editing a program while another program is running. The teach pendant is not in control of the robot during this time. Personnel could be injured or equipment damaged,

During background editing, you **can**

- Create and delete programs.
- Add new program instructions.
- Add new motion instructions.

The position recorded will be the current position of the robot.

- If the robot is currently executing a motion instruction in another program, the robot position at the time you add the motion instruction will be the recorded position.
- If the robot is not executing a motion instruction in another program, the current robot position will be the recorded position.

- Modify existing program instructions.

You cannot enable motion from the background program. You cannot move the robot unless the teach pendant is enabled.

If you add motion instructions during background program editing, you must remember to touch up the positions using TOUCHUP in the foreground, before you run the program.

Use Procedure 5-3 to modify a program in the background.

5.3.1 Background Edit Process Flow

Figure 5-6 and Figure 5-7 show how the Background Edit process flows.

Figure 5-6. Background Edit Process

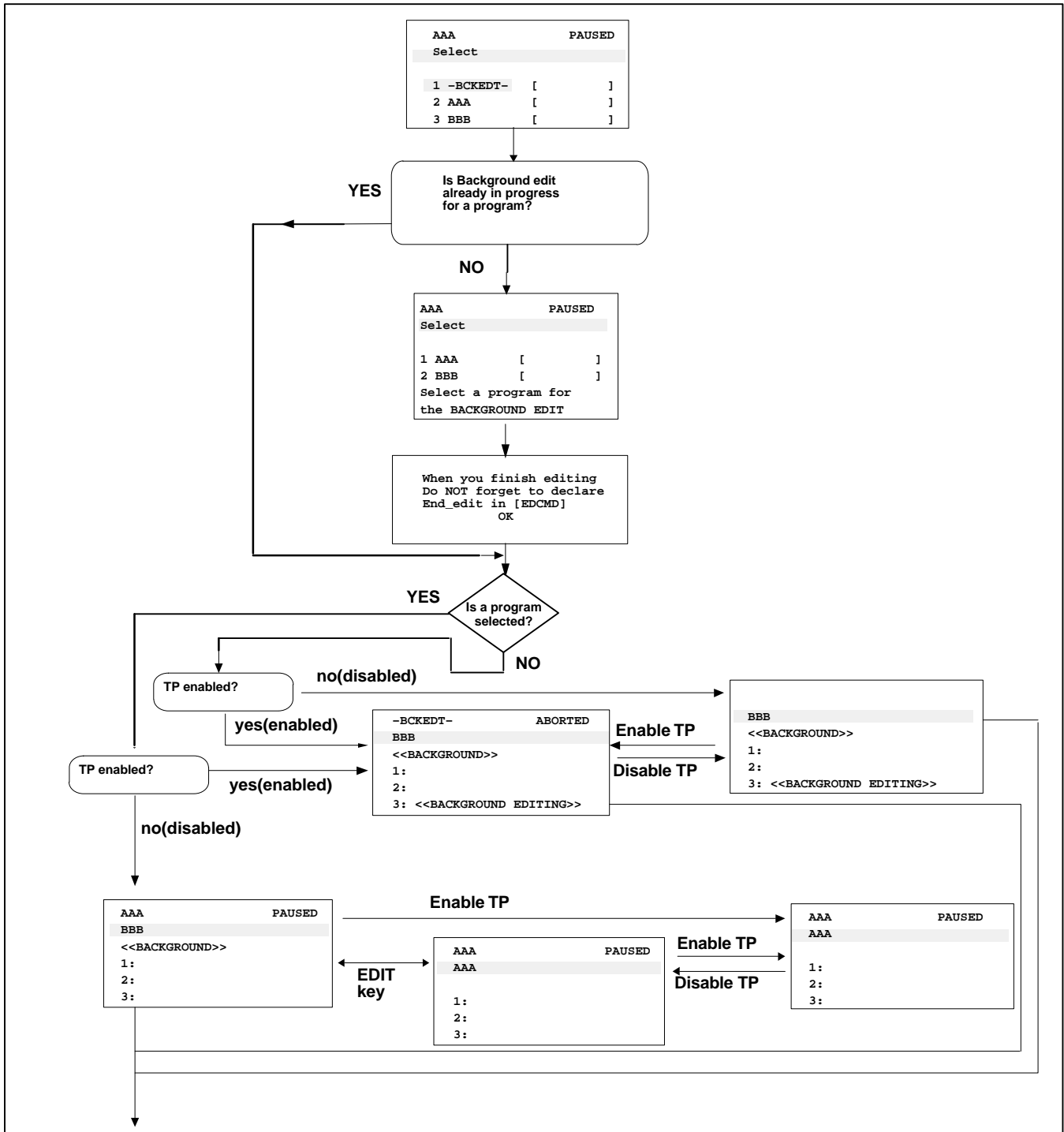
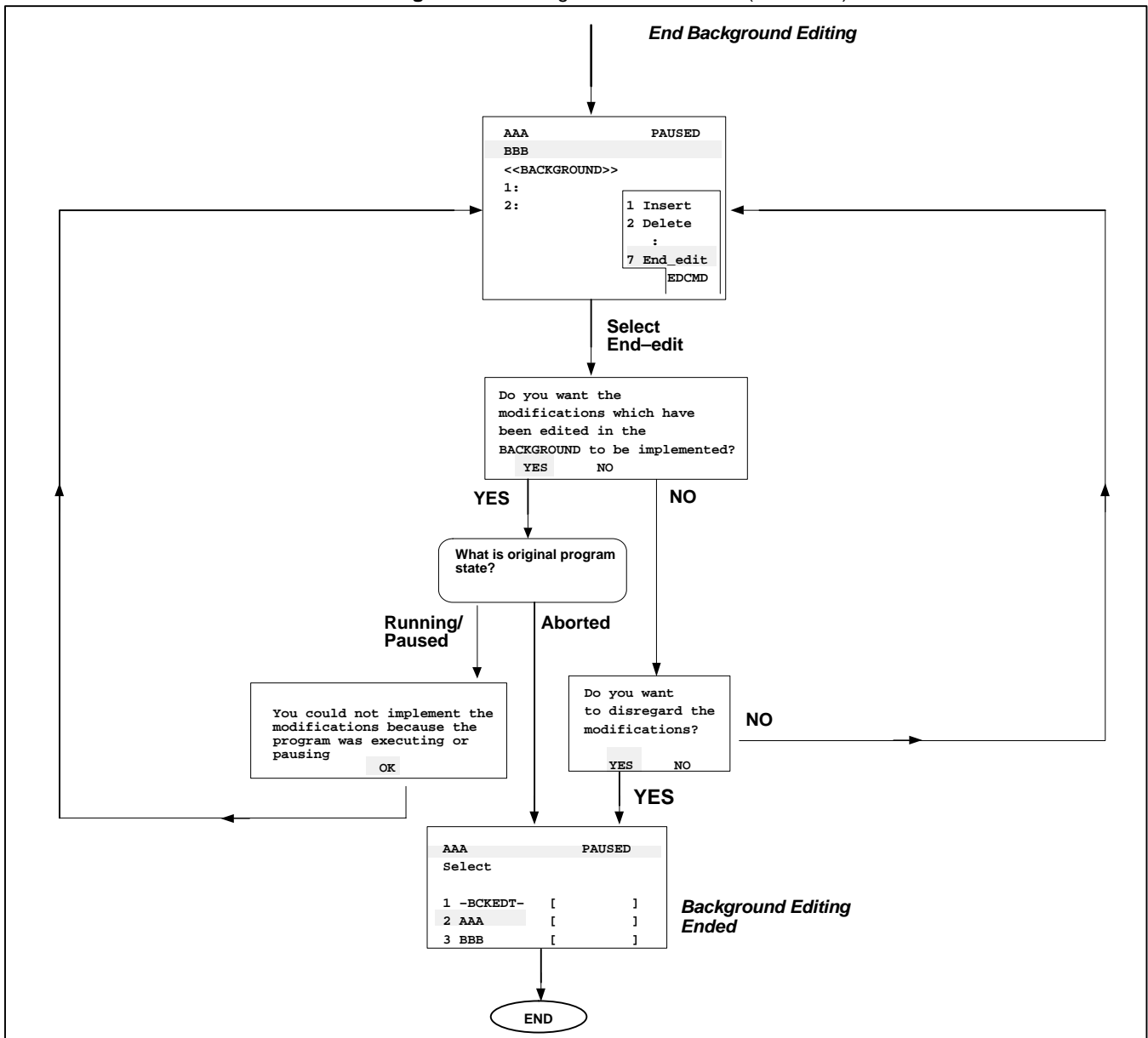


Figure 5-7. Background Edit Process (continued)



Procedure 5-3 Modifying a Program in the Background

- Condition**
- All personnel and unnecessary equipment are out of the workcell.
 - The program has been created and all detail information has been set correctly. (Procedure 5-1)
 - Make sure the \$BACKGROUND system variable is set to TRUE.

- Step** 1 Disable the teach pendant.

NOTE If the teach pendant is enabled when you perform this procedure, the program you select for background edit will be executed instead.

- 2 Press SELECT. You will see a screen similar to the following.

Select		JOINT 10%	
		287746 Bytes free	1/3
No.	Program name	Comment	
1	-BCKEDT-	[]
2	COND	[]
3	MAIN	[]
4	MSG	[]

- 3 Select the special program used for background editing. The name of this program is -BCKEDT-.
- **If a program is currently running in the background**, you will automatically be taken back to the background editing session. Go to Step 7.
 - **If a program is not already running in the background**, you must select a program to edit in the background. You will see a screen similar to the following.

Select		JOINT 10%	
		287746 Bytes free	1/3
No.	Program name	Comment	
1	COND	[]
2	MAIN	[]
3	MSG	[]
Select a program for the BACKGROUND EDIT.			
[TYPE] CREATE DELETE MONITOR [ATTR]>			

- 4 Move the cursor to the name of the program you want to edit.

When you finish editing
DO NOT forget to declare
End_edit in [EDCMD].

OK

EDIT Function Key

External Start Signal

During Operation

External Program Select

Do you want the modifications
which have been edited in the
background to be implemented?

YES

NO

You could not implement the
modifications because the
program was executing or
pausing.

OK

5 Press ENTER.

6 You will see a confirmation message. Press ENTER.
“<<BACKGROUND>>” will be displayed at the beginning of the
program.

7 Modify the program. Refer to Procedure 5-2 .

NOTE Any modifications you make to the background program do not
take effect until you end background editing.

8 **To toggle between two programs**, one in the foreground and one in
the background, press EDIT. If no program is selected in the
foreground, the special program (BCKEDT) is brought to the
foreground.

NOTE You cannot edit two or more programs in the background at the
same time. To edit another program in the background, you must first end
the background editing of the first program by selecting End_edit. Then
restart background editing. Disable the teach pendant. Press the EDIT key
or re-select –BCKEDT– on the program Select screen.

If an external start signal is received during background editing, the
program selected in the foreground is started.

**If the program you selected for background edit is run during
automatic operation**, or called as a subprogram, the original program is
executed (instead of the program which you changed in the background).

If you select a program with an external program selection function,
(such as PNS) during background editing, background editing will
continue normally.

9 **When you are finished editing the program in the background**,
end the background editing session:

a Press F5, [EDCMD].

b Select End_edit.

10 Save the changes.

- **If you want to save the changes you made**, move the cursor to
YES and press ENTER. You will be returned to the program
SELECT screen and <<BACKGROUND>> will no longer be
displayed at the beginning of the program.
- **If you do not want to save the changes you made**, move the
cursor to NO and press ENTER. You will be given the option to
disregard the changes or be returned to the current background edit
session.

NOTE You cannot implement the changes you made if the
currently selected program is running or paused. You must first
select OK and press ENTER before you can save the program.
You will be returned to the background editing session.

5.3.2 Troubleshooting Background Edit

When using Background Edit you might experience one or more of the problems mention in Table 5-2. Use the Cause and Remedy information in Table 5-2 to eliminate any problems that might occur during normal operation.

Table 5-2. Troubleshoot Background Edit – Problem Cause and Remedy

Problem	Cause	Remedy
TPIF-005 Program is not selected was displayed when you pressed the EDIT key.	A program is not selected in the foreground.	There must be two programs selected (one in the foreground and one in the background) in order to use the EDIT key to toggle between them.
TPIF-053 Not editing background program was displayed when you pressed the EDIT key.	A program is not selected in the background.	
You cannot start background edit for the selected program.	There is not enough memory available to copy the selected program for editing.	The amount of available memory must be larger than the size of the selected program to start background editing.
TPIF-054 Could not end editing or MEMO-126 No more available memory Not enough memory available to save the changes you made during background editing.	When background editing ends, a backup copy of the original program is created. The original program is then modified. These two programs require more memory than is currently available.	The amount of memory must be larger than the original program and the program copied for background editing, in order to save any changes you made during the background editing session.
TPIF-055 Could not recover original program The original program is corrupt and cannot be recovered.	Power to the robot was turned off, then back on during background editing.	You must recover the backup version. Check the original program. Then test the program continuously to eliminate the possibility of any errors occurring. Refer to Chapter 7.
The robot stops and the following message is displayed: Program was executing or ...	You tried to run the original program before ending the background editing session.	Do not run the original program until you end (End_edit) background editing.
The robot stops, and the following error messages are displayed: SYST-011 Failed to run task or MEMO-004 Specified program is in use		
The robot stops, and the following error messages are displayed: SYST-011 Failed to run task or MEMO-008 Specified line no. not exist		
TPIF-054 Could not end editing or TPIF-008 Memory protect violation	The original program is write protected.	You cannot end background editing. First change the write protection on the original program. Then edit the program in the background.
After you abort a subprogram, the status line continues to indicate the name of the subprogram.	If a main program is selected in the foreground.	The status line indicates the execution state of the selected program.
TPIF-104 Teach Pendant is disabled	The teach pendant is disabled and you are trying to create or delete a program.	Select the program from the Program SELECT screen. The background editing screen will then be displayed.

Table 5-2. (Cont'd) Troubleshoot Background Edit – Problem Cause and Remedy

Problem	Cause	Remedy
The status of a running subprogram, that was executed from the BCKEDT program, changed to ABORTED when you disabled the teach pendant. The program list screen was displayed.	You selected the special program (BCKEDT) for background editing, while the teach pendant was enabled. You then disabled the teach pendant before the program completed.	If you select the special program for background editing while the teach pendant is enabled, do not disable the teach pendant until the program completes.
A program status changes from RUNNING to ABORTED.		
This program is being edited	The special program (BCKEDT) cannot be loaded from a floppy disk if there is already a program for which completed editing is being held in memory.	End background editing (End_edit) before you attempt to load the special program (BCKEDT) from a floppy disk.

6 PROGRAM ELEMENTS

6 PROGRAM ELEMENTS

A program element is a component of a program. A *palletizing* program is a series of program elements selected and organized to perform a palletizing application.

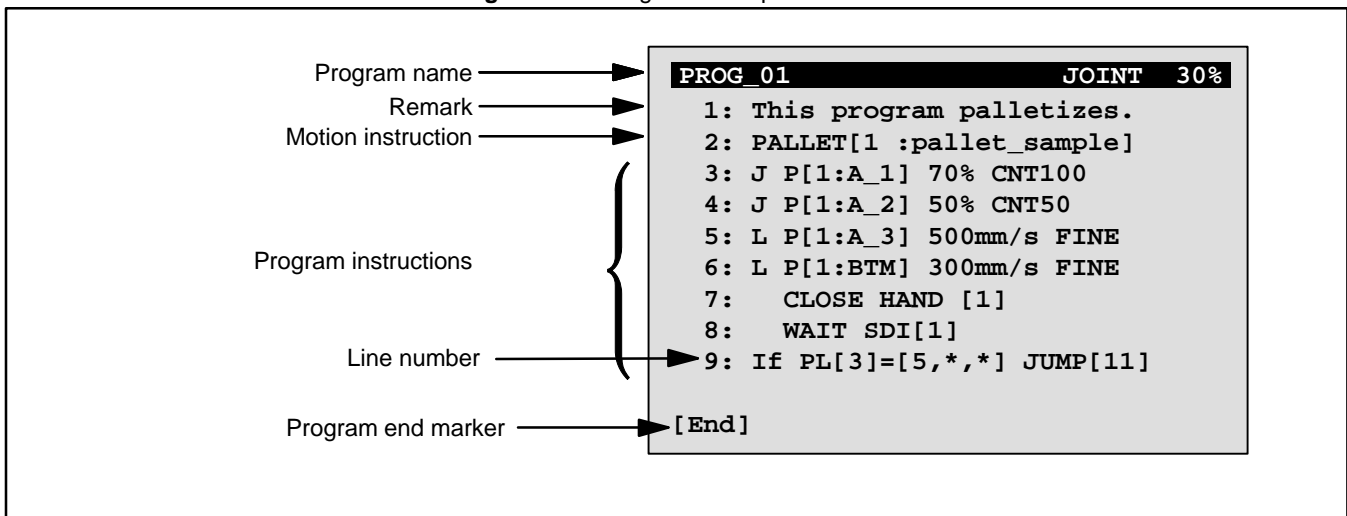
Topics In This Chapter		Page
Program Header Information	Program header information is specific information that identifies and classifies the program.	6-5
	<ul style="list-style-type: none"> ● Creation Date ● Modification Date ● Copy Source ● Positions and Program Size ● Program Name ● Sub Type ● Program Comment ● Group Mask ● Write Protection ● Ignore Pause 	<ul style="list-style-type: none"> 6-5 6-5 6-5 6-6 6-6 6-7 6-7 6-8 6-8 6-9
Line Number and Program End Marker	A line number is inserted automatically next to each instruction you add to a program.	6-10
Motion Instruction	A motion instruction directs the robot to move to a specific location in the workcell using a specified speed and moving in a specified way.	6-11
	<ul style="list-style-type: none"> ● Motion Type ● Positional Information ● Frame Number of Positional Data ● Speed ● Termination Type ● Speed ● Termination Type ● Motion Options ● AccuPath ● Correspondence between Teach Pendant Program Motion and KAREL Program Motion 	<ul style="list-style-type: none"> 6-12 6-20 6-21 6-23 6-32 6-23 6-32 6-34 6-47 6-57
IntelliTrak	The IntelliTrak function improves robot path accuracy.	6-58
Palletizing Instructions	Palletizing instructions tell the robot when and how to palletize.	6-60
	<ul style="list-style-type: none"> ● PALLETIZING-B Instruction ● PALLETIZING-BX Instruction ● PALLETIZING-E Instruction ● PALLETIZING-EX Instruction ● PALLETIZING-END Instruction 	<ul style="list-style-type: none"> 6-62 C-21 6-75 6-82 6-90
Pallet Register Instructions	A pallet register stores layer, row, and column information for palletizing.	6-91
Register Instructions	A register stores one number.	6-93
Position Register Instructions	A position register stores positional information (x, y, z, w, p, r, configuration)	6-96
	<ul style="list-style-type: none"> ● PR[x] Position Register Instructions ● PR[i,j] Position Register Element Instructions 	<ul style="list-style-type: none"> 6-96 6-97
Input/Output Instructions	Input/Output, or I/O, instructions allow the program to turn on and off output signals and receive input signals.	6-100
	<ul style="list-style-type: none"> ● Digital Input and Output Instructions ● Robot Digital Input and Output Instructions ● Analog Input and Output Instructions ● Group Input and Output Instructions 	<ul style="list-style-type: none"> 6-100 6-101 6-103 6-104

Topics In This Chapter		Page
Branching Instructions	Branching instructions cause the program to branch, or jump, from one place in a program to another	6-105
	<ul style="list-style-type: none"> ● Label Definition Instruction ● Unconditional Branching Instructions ● Conditional Branching Instructions 	6-105 6-105 6-106
Wait Instructions	Wait instructions delay program execution for a specified time or until a specified condition is true.	6-109
Miscellaneous Instructions	There are miscellaneous instructions for production control, user alarms, timer setting, speed override, program remarks, message handling, and parameter setting	6-112
	<ul style="list-style-type: none"> ● RSR Enable/Disable Instruction ● User Alarm Instruction ● Timer Instruction ● OVERRIDE Instruction ● Remark Instruction ● Message Instruction ● Parameter Name Instruction ● Maximum Speed Instruction 	6-112 6-112 6-113 6-113 6-113 6-113 6-114 6-118
Skip Instruction	The skip instruction sets the conditions for executing robot motion when using the skip motion option in a motion instruction	6-119
Offset Instruction	Offset instructions specify positional offset information or the frames used for positional information	6-121
Tool Offset Condition Instruction	A tool offset condition instruction specifies the offset condition used in a tool offset instruction	6-124
Multiple Control Instructions	Multiple control instructions are used for multi-tasking.	6-125
Macro Command Instruction	The macro command instruction specifies the macro command to be executed when the program is run.	6-126
Program Control Instructions	Program control instructions direct program execution	6-127
	<ul style="list-style-type: none"> ● PAUSE Instruction ● ABORT Instruction ● Error Program Instruction ● Resume Program Instruction 	6-127 6-127 6-128 6-128
Sensor Instructions	Sensor instructions send and receive information from external sensors through the RS-232-C serial port	6-129
Motion Group Instructions	Motion group instructions allow you to program multiple motion groups independently of each other	6-131
	<ul style="list-style-type: none"> ● Independent Motion Group Instructions ● Simultaneous Motion Group Instructions ● Defining Motion Group Instructions 	6-131 6-131 6-132
Position Register Look-Ahead Instructions	While the robot is executing a program, it reads the lines ahead of the line currently being executed (look-ahead execution).	6-134
Condition Monitor Instructions	The condition monitor function monitors the condition of an I/O signal, register value, or alarm status during teach pendant program execution.	6-135

Topics In This Chapter		Page
Payload Instruction	For some applications, you might need to adjust the payload several times within your teach pendant program	6-137
Collision Guard Instructions	You can use the Collision Guard instructions to control Collision Guard during programmed motion	6-139

Figure 6-1 shows some of the program elements of a typical palletizing application program.

Figure 6-1. Program Example



A palletizing program consists of the following program elements:

- Program header information, which includes a program name, comment, group mask, program type, and write protection setting
- Line numbers, assigned to each program instruction
- Motion instructions, which include commands that tell the robot where and how to move
- Program instructions, which include
 - HandlingTool instructions to palletize.
 - Register instructions to store numerical program information.
 - Position register instructions to store program positional information.
 - Input/Output (I/O) instructions to send signals to and receive signals from equipment in the workcell.
 - Branching instructions to control the direction and order of program flow.

- Wait instructions to delay program execution.
- Skip instructions to move the robot until a signal is received. After the signal is received, stop and branch to the specified statement.
- Offset and tool offset instructions to offset positional information.
- Multiple control instructions to control different motion groups.
- Macro command instructions to perform specific functions.
- Program control instructions to direct program execution.
- Miscellaneous instructions to allow functions such as writing messages to the screen.
- Sensor instructions to control sensors.
- Motion group instructions to use motion groups.
- Position register look-ahead instructions to control motion execution.
- Condition monitor instructions to monitor I/O, register, and alarm conditions during program execution.
- Payload instructions to set the appropriate payload schedule.
- Collision Guard instructions to use Collision Guard in a program.
- Remarks to annotate the program
- Program end marker, indicating that there are no more instructions in the program

6.1 PROGRAM HEADER INFORMATION

Program header information is specific information that identifies and classifies the program. Program header information consists of

- Creation date
- Modification date
- Copy source
- Whether program contains positions and program size
- Program name
- Sub type
- Program comment
- Group mask
- Write protection
- Ignore pause

Program header information is displayed only the first time you create a program. If you want to view this information again, you must display it by choosing the SELECT menu and pressing the DETAIL key. See the following screen for an example.

```

Program Detail                                JOINT 10%
                                                    1/5
Creation date:                                02-Jan-9x
Modification Date:                            02-Jan-9x
Copy source: [                                ]
Positions: TRUE Size                          17 byte
 1 Program Name [ PROG742 ]
 2 Sub Type: [ NONE ]
 3 Comment: [ ]
 4 Group mask: [1,*,*,*,*]
 5 Write protect: [ ON ]

```

The following sections contain details on each type of program header information.

6.1.1 Creation Date

Creation date is the date on which the program name was created.

6.1.2 Modification Date

Modification date is the date, according to the calendar in the controller, when the file was last pulled into the Editor. This information can be displayed using the [ATTR] function key on the SELECT menu.

6.1.3 Copy Source

Copy source is the name of the file from which the file was copied. This field is empty if the file is an original file. This information can be displayed using the [ATTR] function key on the SELECT menu.

6.1.4 Positions and Program Size

Positions indicates whether the program contains recorded robot positions. When you first create a program, positions is always set to FALSE. Size is the size of the program in bytes. The program size can be displayed using the [ATTR] function key on the SELECT menu.

6.1.5 Program Name

A program name identifies the program. When you create a new program, you must give it a unique program name. The program name differentiates the programs stored on the controller.

Length

The name can be from one to eight characters long.

Available Characters

Letters: A through Z. A program name must start with a letter.

Numbers: 0 through 9; a program name cannot start with a number.

Symbols: _ (underscore) only; **do not use** @ (at), * (asterisk), or space.

Content

The name should be descriptive and it should tell you what the program does.

NOTE If you are writing a program for production operation using RSR or PNS, name the program as follows:

- An **RSR** program must be **RSRnnnn**, where **nnnn** is a four-digit number. For example, if you want your program numbered 23, you would enter RSR0023.
- A **PNS** program must be **PNSnnnn**, where **nnnn** is a four-digit number. For example, if you want your program numbered 23, you would enter PNS0023.

NOTE Refer to Section 3.9 for more information on setting up RSR and PNS programs to run in production.

6.1.6 Sub Type

Sub type identifies the kind of program you want to write. These are:

- None
- Macro
- Cond

None

If you select none, the program will not have a sub type. This means that you can include any instructions in your teach pendant program.

Macro

A macro program can contain any instruction and function as a normal program. However, only macro programs can be set up to be executed from one of the following:

- Teach pendant keys
- Manual Functions menu

Refer to Chapter 3 for more information about macros. Macro programs can also be called by a program when the MACRO instruction is used. Refer to Section 6.17.

Cond

A “ch” program, has a Cond sub type. Refer to Section 10.18 for more information on the condition monitor function.

NOTE If you have the condition monitor function option, you can use a “ch” program, which has a Cond sub type. Refer to Section 10.18 for more information on the condition monitor function.

6.1.7 Program Comment

When you create a new program, you can also add a program comment to the name. A program comment includes additional information that you want to further identify the program. This information can be displayed using the [ATTR] function key on the SELECT menu, and also from the DETAIL function key.

Length

One to 16 characters

Available Characters

Letters: a through z and A through Z

Numbers: 0 through 9

Symbols: _ (underscore), @ (at), * (asterisk)

Blank spaces

Punctuation: ; (semicolon), : (colon), ” (quotation marks), () (left and right parentheses)

Content

Should be descriptive, providing additional information.

6.1.8 Group Mask

When you create a program, you define the group mask that identifies the group of axes, or motion group, that the program will control. Motion groups define different groups of axes that can be used for independent pieces of equipment, positioning tables, and other axes.

There are three motion groups available. The controller can operate a maximum of 16 axes; however, only nine axes can belong to a single motion group.

NOTE Multiple Motion (MM) groups must be set up before they can be used.

If a system has only one motion group, the default motion group is 1. An asterisk indicates the group is not used. You can specify a program to use all three motion groups, but only two motion groups can perform Cartesian interpolated motion within a single program. If you disable all groups, you cannot add motion instructions to your program.

When you add a motion instruction that has a motion type, the following appears on the screen:

Group mask: 1, *, *, *, *

You cannot change the group mask after you have added motion instructions to your program.

6.1.9 Write Protection

Write protection allows you to specify whether the program can be modified.

- **When write protection is set to ON**, you cannot add or modify any element in the program. When you have finished creating a program and are satisfied with how it works, you should set write protection to ON so that you or someone else does not modify it.
- **When write protection is set to OFF**, you can create the program and add or modify any element in the program. By default, write protection is set to OFF.

This information can be displayed using the [ATTR] function key on the SELECT menu.

 **CAUTION**

When write protection is set on, you cannot modify any program header information (program name, sub type, program comment, group mask, ignore pause) except write protection.

6.1.10 Ignore Pause

Ignore pause allows you to specify whether the program will continue to run even when an error occurs, a command is issued (such as pushing EMERGENCY STOP or HOLD), or the teach pendant is enabled. Ignore pause is allowed only in programs that do not have motion groups specified. This means that programs that use ignore pause cannot contain any motion instructions.

**WARNING**

If ignore pause is set to ON, the program MUST NOT issue any motion instructions; otherwise, injury to personnel or equipment can occur.

- **When ignore pause is set to ON**, the program continues to run even when an error occurs, a command is issued, or the teach pendant is enabled. This allows the program to continue any monitoring function, such as monitoring I/O.
- **When ignore pause is set to OFF**, the program pauses when an error occurs, a command is issued, or the teach pendant is enabled.

6.2 LINE NUMBER AND PROGRAM END MARKER

A line number is inserted automatically next to each instruction you add to a program. If you remove an instruction or move an instruction to a new position in the program, the program instructions will be renumbered automatically so that the first line is always line 1, the second line 2, and so forth. You use line numbers to identify which lines to move, remove, and mark when modifying a program.

The program end marker (**[End]**) automatically appears after the last instruction in a program. As new instructions are added, the program end marker moves down on the screen, retaining its position as the last line in the program.

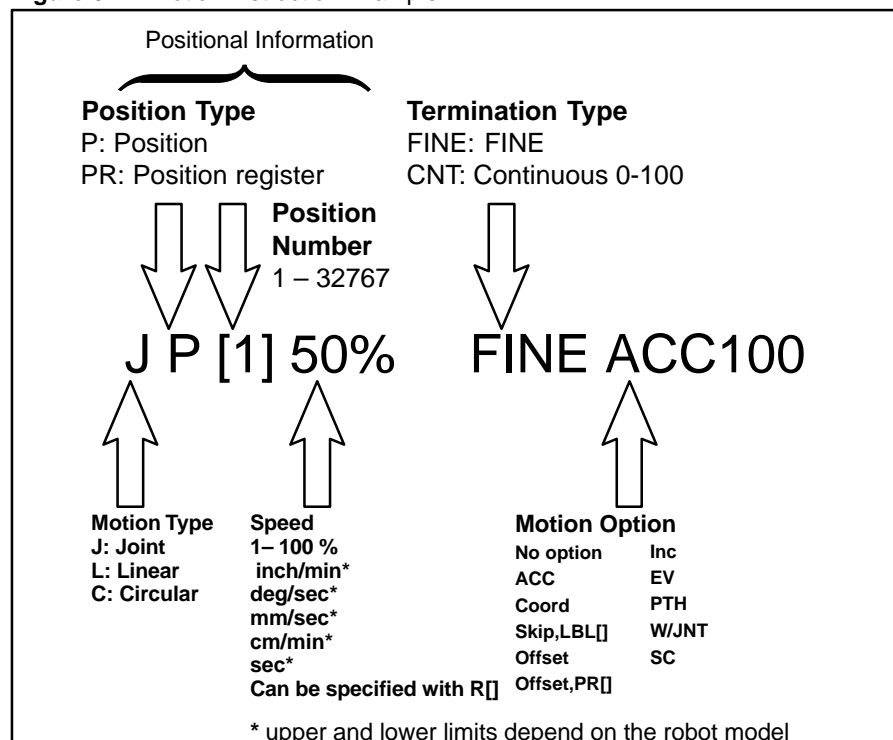
6.3 MOTION INSTRUCTION

A motion instruction directs the robot to move to a specific location in the workcell using a specified speed and moving in a specified way. A motion instruction includes:

- **Motion type** – How the robot moves to the position
- **Positional information** – Where the robot moves
- **Termination type** – How the robot ends the move to the position
- **Speed** – How fast the robot moves to a position
- **Motion options** – Additional commands that perform specific tasks during robot motion.

An example motion instruction is shown in Figure 6-2.

Figure 6-2. Motion Instruction Example



NOTE Wrist Joint (W/JNT) is a Motion Option that only moves only in Linear or Circular.

6.3.1 Motion Type

Motion type defines how the robot will move to the destination position. There are three motion types:

- Joint
- Linear
- Circular

Joint Motion

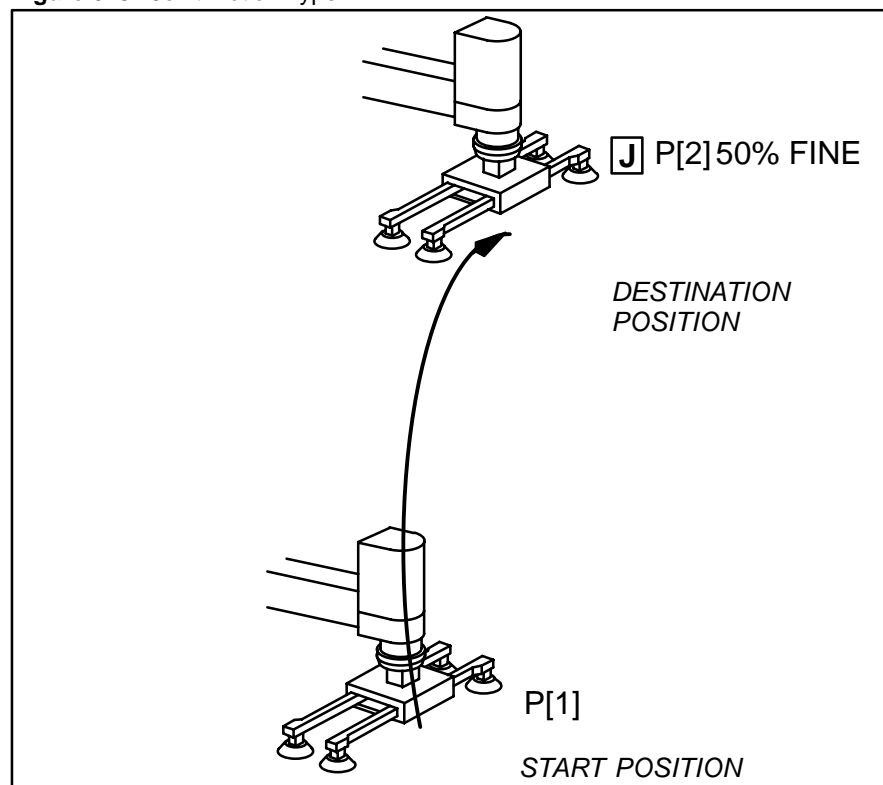
J P[2] 50% FINE

Joint motion

- Causes the robot to move all required axes to the destination position simultaneously. The motion of each axis starts and stops at the same time.
- Is programmed at the destination position.
- Speed is specified as a percentage of the total default speed, or in seconds. The actual speed of the move is dependant on the speed of the slowest axis. Refer to Section 6.3.4.

Figure 6-3 shows an example of joint motion.

Figure 6-3. Joint Motion Type



Linear Motion**L** P[2] 100mm/sec FINE

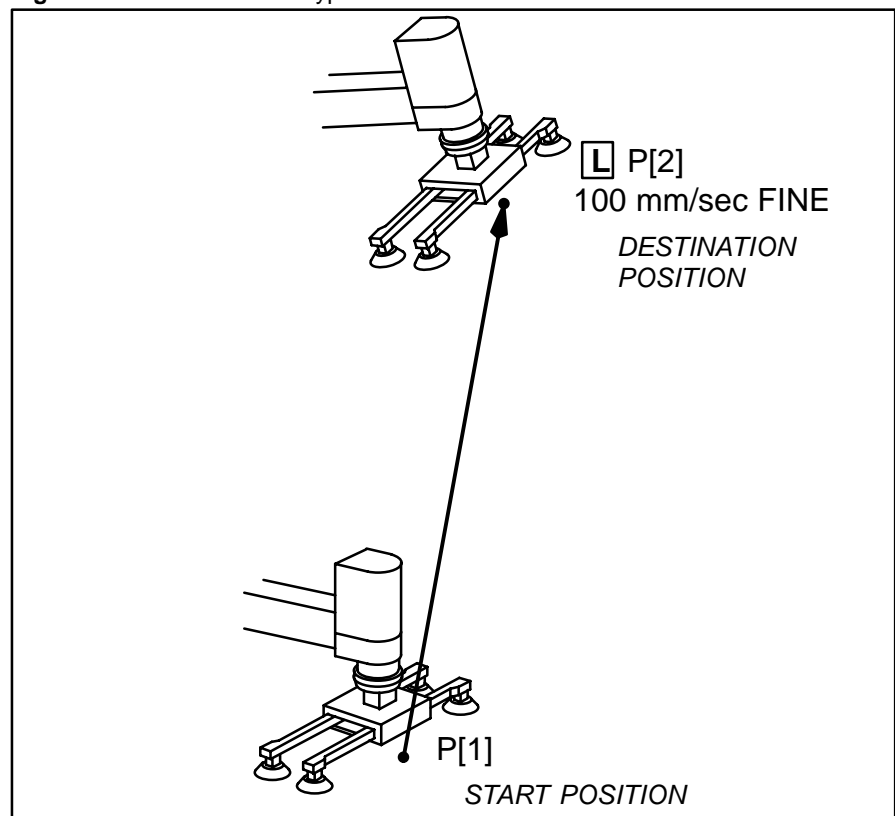
Linear motion

- Causes the robot to move the tool center point in a straight line from the start position to the destination position.
- Is programmed at the destination position.
- Speed is specified in millimeters per second, centimeters per second, inches per minute, degrees per second, or seconds. Refer to Section 6.3.4.

During a linear move, the orientation of the tool changes gradually as the robot moves from the start position to the destination position, depending on how the destination position is programmed.

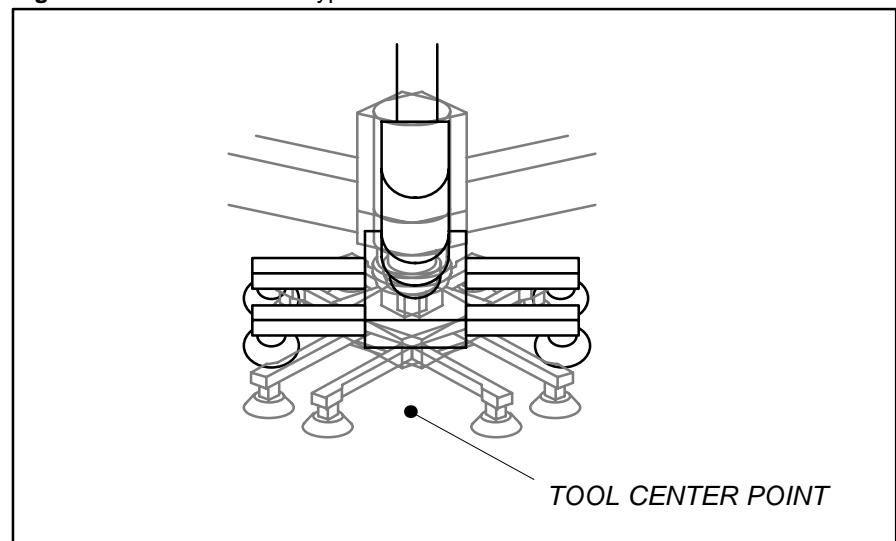
Figure 6-4 shows an example of linear motion.

Figure 6-4. Linear Motion Type



Linear motion type can also be used to rotate tool about the tool center point while maintaining that position. The speed for this type of motion is in degrees per second. Figure 6-5 shows an example of linear motion used for rotation about the tool center point.

Figure 6-5. Linear Motion Type Used to Rotate About the Tool Center Point



Circular Motion

C P[2] P[3] 100mm/sec FINE

Circular motion

- Causes the robot to move the tool center point in an arc from the start position through an intermediate to the destination position.
- Is programmed at the intermediate position.
- Speed is specified in inches per minute, millimeters per second, and centimeters per minute. Refer to Section 6.3.4.

When you add a motion instruction that has circular motion type, the following appears on the screen:

```

C P[2]
P[3] 100 mm/sec FINE
  
```

The first position, P[2] in the example, is the intermediate position. The intermediate position is recorded automatically as the current robot position when you add the motion instruction. The second position, P[3] in the example, is the destination position. You must record the destination position, after you add the circular motion instruction, using the TOUCHUP function key, F5.

If you change an existing point to “C”, that position becomes the “Via”, or intermediate, position.

To program a complete circle, add two circular motion instructions. The circular motion instructions can be added by:

- Inserting a line.
- Returning to DEFAULT.
- Selecting [INST].
- Editing a default instruction to add the circular motion instruction.
- Teaching a point with the current default and then modifying the line to become a circular motion statement.

The following program instructions can be used to create a full circle.

```
J P[1] 100mm/sec FINE
C P[2]
  P[3] 100mm/sec
C P[4]
  P[1] 100mm/sec FINE
```

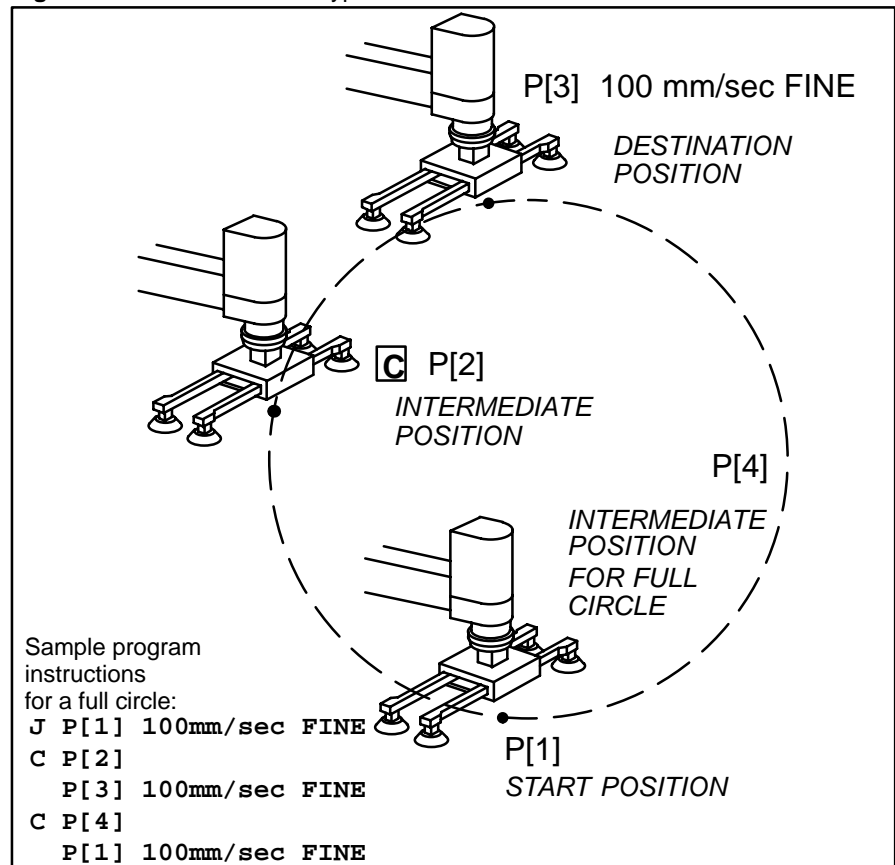
Figure 6-6 shows an example of circular motion.

Circular Orientation Control at Intermediate Via Point

Circular orientation control at the intermediate “via” point ensures that the robot will go through the “via” point at the taught orientation point. Orientation is smoothly inserted between the start, via, and end points.

Figure 6-6 shows an example of circular motion.

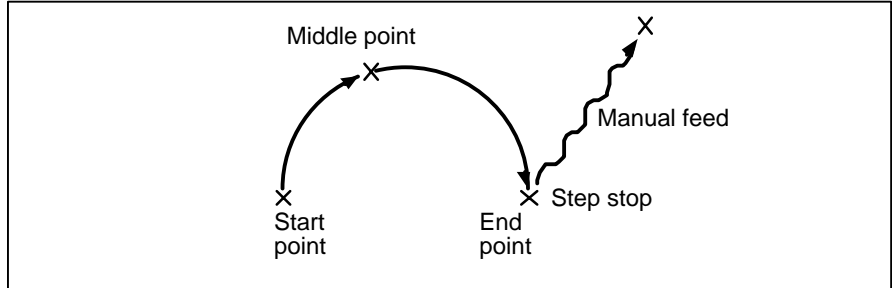
Figure 6-6. Circular Motion Type



Restart of Circular Motion

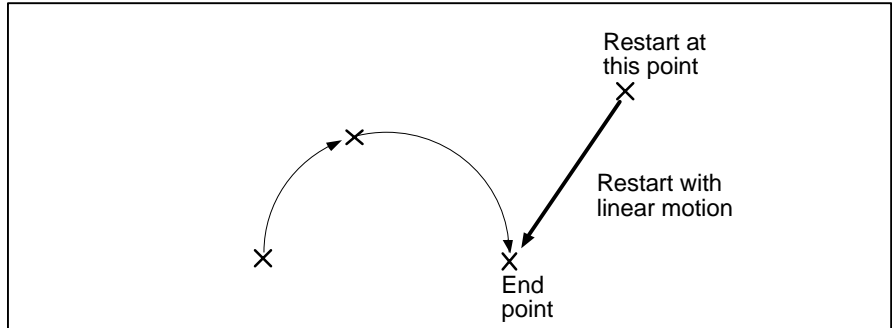
In Figure 6-7 a single-step stop occurs at the destination position of a circular motion instruction. You can then jog the robot.

Figure 6-7. Restart of Circular Motion Instruction



In Figure 6-8 when program execution is restarted after a single-step stop and jogging, the robot moves, using linear motion, to the end point of the previous circular motion.

Figure 6-8. Restart of Circular Motion Instruction



Guidelines for Teaching a Small Circular Arc

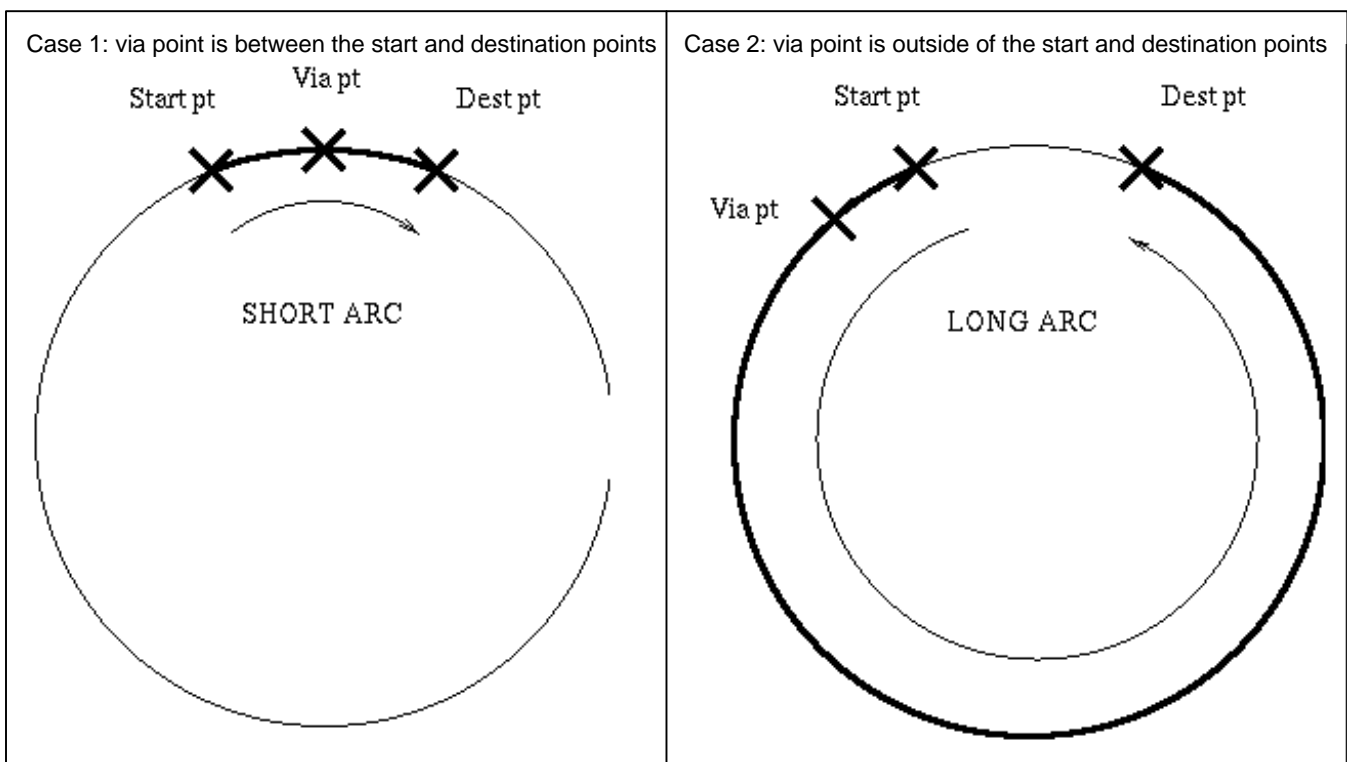
In general, we do not recommend that you teach a very small circular arc with large orientation changes. Even with small orientation changes, it is important to teach circular points correctly to achieve the circular arc you want. The information in this section illustrates the importance of proper location and orientation of the start, via, and destination positions in creating a circular arc.

NOTE If your application requires very small circles, refer to the *SYSTEM R-J2 Controller HandlingTool Software Installation Manual* for information on installing the Small Circle Accuracy option.

Example 1: Effect of Via Point Location

See Figure 6-9 for an illustration of the effect of the via point location.

Figure 6-9. Effect of Via Point Location

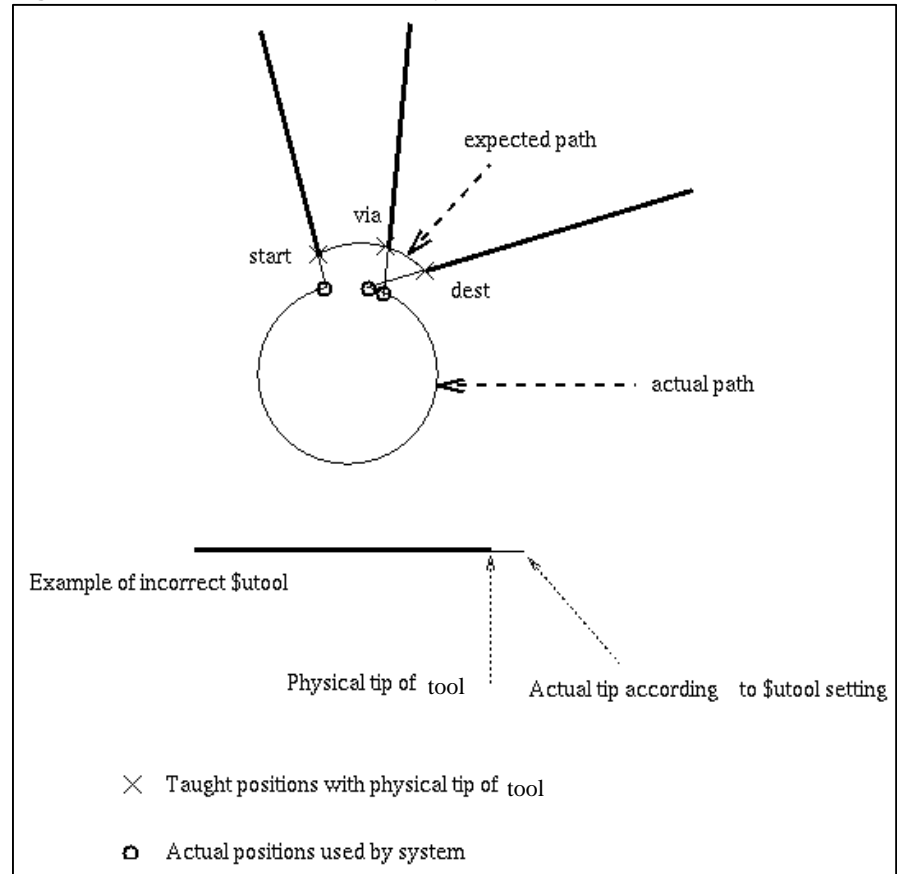


Depending on where the via point is with respect to the start and destination points, the circular arc can be short or long. With a large UTOOL offset and large orientation changes, the robot faceplate movement for a long arc is much greater than the faceplate movement for a short arc.

It is best to teach the via point half-way between the start and destination points. Otherwise, during touchup, the via point could end up on the other side of the arc, which could cause motion other than what you expect.

Example 2: Effect of UTOOL Accuracy

See Figure 6-10 for an illustration of the effect of UTOOL accuracy.

Figure 6-10. Effect of UTOOL Accuracy

If the UTOOL is not accurate, especially in the case of a large tool offset with a small circular arc, the taught path might appear to have the via point between the start and destination points, even though the via point is outside these points. In Figure 6-10, a long arc results instead of the expected short arc.

Example 3: Effect of Via Point Orientation

Use the following guidelines for via point orientation:

- For a small circular arc, do not teach a large orientation change. If you teach a large orientation change, fast orientation motion will occur, even though the Tool Center Point location speed is planned according to the programmed speed.
- It is best to teach the via point location and orientation approximately half-way between the start and destination points. If the via point is too close to the start point compared to the destination point (or vice-versa), when you touch up the via point, the via point could end up on the other side of the arc. In this case, the planned orientation motion will be in a different direction, resulting in a large orientation change.

A software feature, called Large Orientation Detection for Small Circle, can be enabled to detect certain large orientation changes. To enable this feature, set the system variable `$CRCFG.$lgorn_enbl` to TRUE (default is FALSE), turn off controller power, and then turn it back on.

When the Large Orientation Detection for Small Circle feature is enabled, during single step testing, if a large orientation change is detected for small circular moves (with a radius of less than 30 millimeters), the system will slow down automatically, and the warning, “MOTN-319 CRC large orient change” will be displayed.

If the large orientation is what you intended to teach, no further action is required. If you do not want the orientation change, release the SHIFT key or press the HOLD key to pause the motion. Then, reteach the circular points to avoid the large orientation changes.

6.3.2 Positional Information

J **P[1]** 50% FINE

Positional information describes the **location of the tool center point** when a motion instruction is added to a program. Positional information is recorded when the motion instruction is added to the program. Refer to Section 5.2.2 for more information on adding motion instructions.

Positional information is made up of seven components, as shown in Figure 6-11. These components are represented by the position command, P[n].

Figure 6-11. Positional Information

$P[n] = (x, y, z, w, p, r, \text{config})$ <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="text-align: center;">  <p>Location</p> </div> <div style="text-align: center;">  <p>Orientation</p> </div> <div style="text-align: center;">  <p>Configuration</p> </div> </div>

- **Location** components, (x,y,z), describe the three-dimensional location of the position.
- **Orientation** components, (w,p,r), describe rotation about x, rotation about y, and rotation about z.
- The **configuration** component describes the condition of the axes when the robot arrives at the destination position. Orientation of the wrist axes at the destination position remains the same, but the orientation of the other axes might change.

In a motion instruction, positional information is represented as a position command, P[n], or position register, PR[x]. The **n** is the position number. The **x** is the position register number. A position command stores positional information with the motion instruction in the program. A position register stores positional information in a storage location separate from the motion instruction. Refer to Section 6.8.

The position number identifies the position. Position numbers are assigned automatically when a motion instruction is added to a program. The first number assigned is [1], the second [2], and so forth.

If you add a position before an already existing position, the position number is incremented from the last numbered position regardless of its place in the program. You can request that positions be renumbered so that the position numbers are sequential in your program. Refer to Section 5.2.2 for more information.

When you delete positions, all other taught positions keep their current numbers unless you request that they be renumbered.

Positions can also have comments of one to 16 characters. You specify these when you add or modify positional information.

Refer to Chapter 5 for more information on modifying the positions in your program.

6.3.3 Frame Number of Positional Data

The User Frame (UF) and User Tool frame number (UT) are displayed at the top of the Position Detail screen. See the following screen for an example.

P[1] UF:0 UT:1 CONF: N 00					
X	100.000	mm	W	12.555	deg
Y	100.000	mm	P	3.123	deg
Z	100.000	mm	R	.014	deg

These fields indicate the current frame number.

UF: User Frame number

0 = world coordinate

1-5 = normal UFRAME number

F = current \$MNUFRAMENUM

UT: User Tool frame number

0 = tool coordinate at face plate

1-5 = normal UTOOL number

F = current \$MNUTOOLNUM

NOTE These values cannot be modified directly from the teach pendant.

NOTE The position register screen has UF and UT in the same area, and this value is always “F” for both.

Switch Frame Check Type

This function provides an easy way to move forward or backward past a frame change in a teach pendant program. You must set \$FRM_CHKTYYP to move through the frame change correctly.

The system variable \$FRM_CHKTYYP allows you to switch the frame check type. Refer to Table 6-1.

Table 6-1. \$FRM_CHKTYYP Values

\$FRM_CHKTYYP	DESCRIPTION
-1	This setting prohibits FWD or BWD motion between two points that have different frame numbers.
-2	With this setting, the system does not check the frame number at FWD and BWD execution. The system does not change the frame number \$MNUFRAME_NUM, \$MNUTOOL_NUM when you FWD/BWD execute past a frame change.
2	With this setting, the system does not check the frame number at FWD and BWD execution. The system changes the frame number (\$MNUFRAME_NUM, \$MNUTOOL_NUM) when you FWD/BWD execute past a frame change.

NOTE If a different frame is used for circular motion, the robot will not move, even if \$FRM_CHKTYYP is 2 or -2.

Figure 6–12 shows the frame numbers used for the positional data in a program example.

Figure 6–12. Frame Number of Positional Data Example Program

```

1: UTOOL_NUM = 1
2: J P[1] 100% FINE /* P[1] UT=1 */
3: J P[2] 100% FINE /* P[2] UT=2 */

```

Table 6–2 describes how the example program in Figure 6–12 operates, depending on the setting of the \$FRM_CHKTYP system variable.

Table 6–2. Example Program Operation

\$FRM_CHKTYP Value	FWD Execution	BWD Execution
-1	The system posts a book keeping error at line 3.	You manually set UTOOL_NUM = 2 and backward execute from line 3. The system posts book keeping error at line 2.
-2	The system does not post a book keeping error and the robot moves correctly.	You bwd execute from line 3 to line 2. The system does not post a book keeping error.
-3	The system does not post a book keeping error at line 3 and the robot moves correctly. Also, the system automatically changes UTOOL_NUM to 2 at line 3.	You backward execute from line 3 to line 2. The system does not post a book keeping error at line 2. But the the system automatically changes UTOOL_NUM to 1 when backward executing from line 3 to line 2.

6.3.4 Speed

Speed defines how fast the robot moves to a position.

The motion type used determines the units of speed. Depending on the motion type you want, you can specify speed in millimeters per second, centimeters per minute, inches per minute, rotational degrees per second, or seconds.

When a program is running, you can change the speed using the +% and -% keys on the teach pendant. The value ranges from .01% (very fine) to 100 percent of the programmed speed. Programmed speed is the speed specified in the program.

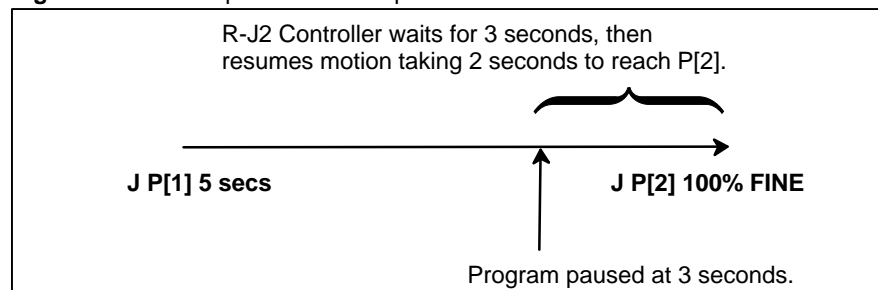
NOTE The programmed speed cannot exceed the capability of the robot. If an invalid programmed speed is entered, an error will occur.

J P[1] **50% FINE**

Joint motion uses

- A percentage (%) of the total default speed. Joint motion speed can have a value of .01% to 100% of the maximum joint speed limit.
- Seconds (sec), the length of time the motion lasts. Seconds can have a value of .1 to 3200 sec. This is used for motion that requires an exact time span. If a program is paused and then resumed during execution of motion that uses seconds, the R-J2 controller will be held in a busy and running state until the portion of time that had been executed elapses again. Then, the robot will complete the motion using the remaining amount of time. See Figure 6-13.

Figure 6-13. Example of the Sec Speed Feature



L P[1] **100mm/sec** FINE

or

C P[1]
P[2] **100 mm/sec** FINE

Linear and circular motions use

- Millimeters per second (mm/sec), with a range of values from 1 to 2000 millimeters per second.
- Centimeters per minute (cm/min), with a range of values from 1 to 12000 centimeters per minute.
- Inches per minute (inch/min), with a range of values from 0.1 to 4724.4 inches per minute.
- Seconds (sec), the length of time the motion lasts. This is used for motion that requires an exact time span. If a program is paused and then resumed during execution of motion that uses seconds, the R-J2 controller will be held in a busy and running state until the portion of time that had been executed elapses again. The robot will then complete the motion using the remaining amount of time. See Figure 6-13.

Rotational control of axes around the tool center point uses rotational degrees per second (deg/sec), with a default range of values from 1 to 90 degrees per second.

Variable Motion Speed

You can specify motion speed by using a register in a motion instruction. The value of the specified register defines motion speed. This is called *variable motion speed*.



WARNING

Before you run a program, make sure you are aware of any register values used to set speed in a motion instruction. Otherwise, unexpected motion could occur resulting in injury to personnel or damage to equipment.

NOTE A program will stop pre-execution of subsequent instructions when it reaches a motion instruction with the register speed type. This ensures the motion instruction uses the register speed type value. See Figure 6-14.

Figure 6-14. Variable Motion Speed Program Execution Example

```
10: R[1] = 100
11: J P[1] R[1]% FINE (program stops pre-execution and takes the value of
12: R[1] = 10          the register in line 10 as the speed.)
13: J P[1] R [1]% FINE (program stops pre-execution and takes the value of
                       the register in line 12 as the speed.)
```

This feature is enabled when the system variable \$RGSPD_PREXE = FALSE. You can disable this feature by setting \$RGSPD_PREXE = TRUE. However, the robot will not be able to move at the speed specified by the register value.

The following examples show various motion type instructions that take their speed value from a register (R[]).

- Joint motion type
J P[2] R[1]% CNT100
- Linear motion type
L P[1] R[2]mm/sec FINE
- Circular motion type
C P[2]
P[3] R[3]cm/min FINE
- Multiple motion group instruction
Independent GP
GP1 L P[4] R[1]mm/sec FINE
GP2 L P[4] R[2]mm/sec FINE
- Palletizing
PALLETIZING-B_1
L PAL_1[A_1] R[4]mm/sec CNT100
L PAL_1[BTM] R[4]mm/sec FINE
L PAL_1[R_1] R[4]mm/sec CNT100

The following features are changed to accommodate variable motion speed:

- **Default motion instructions** include an instruction that uses variable motion speed.
- The **Motion Modify** screen, displayed when you select REPLACE for a motion instruction on the [EDCMD] menu, includes items for specifying variable motion speed.
- Specific **motion speed values** are valid for variable motion speed. Refer to Table 6-3 for valid variable motion speed values. If the specified register value is not a valid speed value (exceeds the speed limit or is out-of-range), an error will occur during the execution of the motion instruction.

Table 6-3. Range of Register Values for Specifying Variable Motion Speed

Unit	Range of Register Values	
%	1 to 100	Integer type
sec	0.1 to 3200.0	Float type (*1)
mm/sec	1 to 2000	Integer type (*2)
cm/min	1 to 12000	Integer type (*2)
inch/min	0.1 to 4724.4	Float type (*3)
deg/sec	1 to 400	Integer type (*4)

*1 : Valid one decimal point.

*2 : The speed limit is the value of \$MRR_GRP.\$SPEEDLIM.

*3 : Valid one decimal point. The limit is the value of \$MRR_GRP.\$SPEEDLIM/25.4 * 60.

*4 : The limit is the value of \$MRR_GRP.\$ROTSPEEDLIM * 180/3.141.

Use Procedure 6-1 to change the motion speed from a specific motion speed value to a variable (register) speed value.

Procedure 6-1 Changing Motion Speed (from Speed Value to Register)

Condition

- You are currently editing a teach pendant program that contains motion instructions.

Change from the Speed Value to a Register

- 1 Move the cursor to the speed value of a motion instruction. See the following screen for an example.

```

RSR0001                                10 %
                                         1/2
1: J P[1] 100% FINE
[End]

Enter value
REGISTER                                [CHOICE]

```

- 2 Press the F1, REGISTER. You will see a screen similar to the following.

```

RSR0001                                10 %
                                         1/2
1: J P[1] R[ ] % FINE
[End]

Enter value
SPEED  DIRECT  INDIRECT [CHOICE]

```

Specify the Speed Value

- 3 Type the register number that contains the speed value by pressing one of the following:
 - F1, SPEED – Allows you to change the speed value back to a specific value and cancel the register variable motion speed.
 - F2, DIRECT – Enter the register number.
 - F3, INDIRECT – Enter the number of a register that contains the register number.

If you press F2, DIRECT, or F3, INDIRECT, you will be prompted to enter a register value.

You will see a screen similar to the following.

```

RSR0001                                10 %
                                         1/2
1: J P[1] R[2] % FINE
[End]

                                         [CHOICE]

```

Input value

Use Procedure 6-2 to change the motion speed from a variable (register) speed value to a specific motion speed value.

Procedure 6-2 Changing Motion Speed (from Register to Speed Value)

Condition ■ You are currently editing a teach pendant program that contains motion instructions.

Change to the Speed Value from a Register

1 Move the cursor to the speed value of a motion instruction. See the following screen for an example.

```

RSR0001                                10 %
                                         1/2
1: J P[1] R[22] % FINE
[End]

Enter value
SPEED  DIRECT  INDIRECT [CHOICE]

```

2 Press F1, SPEED. See the following screen for an example.

```

RSR0001                                10 %
                                         1/2
1: J P[1] [ ] % FINE
[End]

Enter value
REGISTER                                [CHOICE]

```

Specify the Speed Value

3 You will be prompted to enter a speed value, as follows:

Input value

4 Type a speed value and press ENTER. See the following screen for an example.

```

RSR0001                                10 %
                                         1/2
1: J P[1] 20% FINE
[End]

[CHOICE]

```

Use Procedure 6-2 to replace speed values using the Motion Modify screen in [EDCMD] REPLACE.

Procedure 6-3 Replacing Speed Values (using Motion Modify in [EDCMD] REPLACE)

- Condition**
- You are currently editing a teach pendant program that contains motion instructions.

```

RSR0001                                10 %
                                         1/4
1: L P[1] 20.0sec FINE
2: L P[2] 500mm sec FINE
3: L P[3] R[1]mm sec FINE
[End]

[ INST ]                                [ EDCMD ]

```

- Step**
- Move the cursor to the line number of the instruction in which you want to replace speed values and press F5, [EDCMD]. See the following screen for an example.

```

RSR0001                                10 %
                                         1/4
1: L P[1] 20.0sec FINE
2: L P[2] 500mm sec FINE
3: L P[3] R[1]mm sec FINE
                                         +-----+
                                         | 1 Insert |
                                         |   :     |
                                         | 5 Replace |
                                         |   :     |
                                         +-----+
[ INST ]                                | EDCMD |

```

- Select Replace. See the following screen for an example.

```

Select Replace Menu
1 Register          5 Motion modify
2 Call END         6
3 I O              7
4 JMP LBL          8
RSR0001                                10 %
                                         1/4
1: L P[1] 20.0sec FINE
2: L P[2] 500mm sec FINE
3: L P[3] R[1]mm sec FINE
Select item

```

3 Select Motion modify. See the following screen for an example.

```

Modify motion menu
1 Replace speed      5
2 Replace term      6
3 Insert option     7
4 Remove option     8
-----
RSR0001                                10 %
                                           1/4
1: L P[1] 20.0sec FINE
2: L P[2] 500mm sec FINE
3: L P[3] R[1]mm sec FINE

Select item

```

4 Select Replace speed. See the following screen for an example.

```

Select interpolate
1 Unspecified type  5
2 Joint             6
3 Linear           7
4 Circular          8
-----
RSR0001                                10 %
                                           1/4
1: L P[1] 20.0sec FINE
2: L P[2] 500mm sec FINE
3: L P[3] R[1]mm sec FINE

Select source type

```

5 Specify the **interpolation type (motion type)** of the motion instruction you want to search for:

- Unspecified type – searches for joint, linear, and circular motion instructions
- Joint – searches for joint motion instructions only
- Linear – searches for linear motion instructions only
- Circular – searches for circular motion instructions only

See the following screen for an example.

```

Speed type menu
1 All type          5
2 Speed value      6
3 R[ ]             7
4 R[R[ ]]          8
-----
RSR0001                                10 %
                                           1/4
1: L P[1] 20.0sec FINE
2: L P[2] 500mm sec FINE
3: L P[3] R[1]mm sec FINE

Select source speed type

```

6 Specify the **speed type** of the motion instruction you want to search for:

- All type – searches for motion instructions that use a speed value, a variable (register) value, or an indirect variable (register) speed value.
- Speed value – searches only for motion instructions that use a speed value.
- R[] – searches only for motion instructions that use a variable (register) speed value.
- R[R[]] – searches only for motion instructions that use an indirect variable (register) speed value.

See the following screen for an example.

Select motion item	
1 %	5 deg sec
2 mm sec	6 sec
3 cm min	7
4 inch min	8
RSR0001	10 %
	1/4
1:	L P[1] 20.0sec FINE
2:	L P[2] 500mm sec FINE
3:	L P[3] R[1]mm sec FINE
Select destination unit	

7 Specify the **units** of the replacement motion instruction:

- %
- mm/sec
- cm/min
- inches/min
- deg/sec
- sec

See the following screen for an example.

Select motion item	
1 Speed value	5
2 R[]	6
3 R[R[]]	7
4	8
RSR0001	10 %
	1/4
1:	L P[1] 20.0sec FINE
2:	L P[2] 500mm sec FINE
3:	L P[3] R[1]mm sec FINE
Select destination speed type	

8 Select the **speed type** of the replacement motion instruction:

- Speed value – changes the speed of the found (searched) motion instruction to a speed value.
- R[] – changes the speed of the found (searched) motion instruction to a variable (register) speed value.
- R[R[]] – changes the speed of the found (searched) motion instruction to an indirect variable (register) speed value.

See the following screen for an example.

```

RSR0001                                10 %
                                         1/4
1: L P[1] 20.0sec FINE
2: L P[2] 500mm sec FINE
3: L P[3] R[1]mm sec FINE
[End]

Enter register number:

```

9 If you selected R[] or R[R[]], type a register number. See the following screen for an example.

```

RSR0001                                10 %
                                         1/4
1: L P[1] 20.0sec FINE
2: L P[2] 500mm sec FINE
3: L P[3] R[1]mm sec FINE
[End]

Modify OK ?
          ALL      YES      NEXT      EXIT

```

10 Select how you want the found motion instruction to be replaced:

- F2, ALL – changes all found motion instructions below the current line to the specified speed type and value.
- F3, YES – changes only the found motion instruction on the current line to the specified speed type and value.
- F4, NEXT – skips the found motion instruction on the current line and searches the next motion instruction.
- F5, EXIT – ends the motion modify operation.

See the following screen for an example.

```

RSR0001                                10 %
                                         1/4
1: L P[1] R[3]sec FINE
2: L P[2] 500mm sec FINE
3: L P[3] R[1]mm sec FINE
[End]

Search next item?
                      NEXT      EXIT

```

11 Continue searching and replacing as desired.**12** When you are finished searching and replacing, press F5, EXIT.

6.3.5 Termination Type

Termination type defines how the robot ends the move in the motion instruction. There are three termination types:

- Fine
- Continuous
- Corner distance – available only if you have the AccuPath option

The fine and continuous termination types are described in this section. Refer to Section 6.3.7 for information on the corner distance termination type.

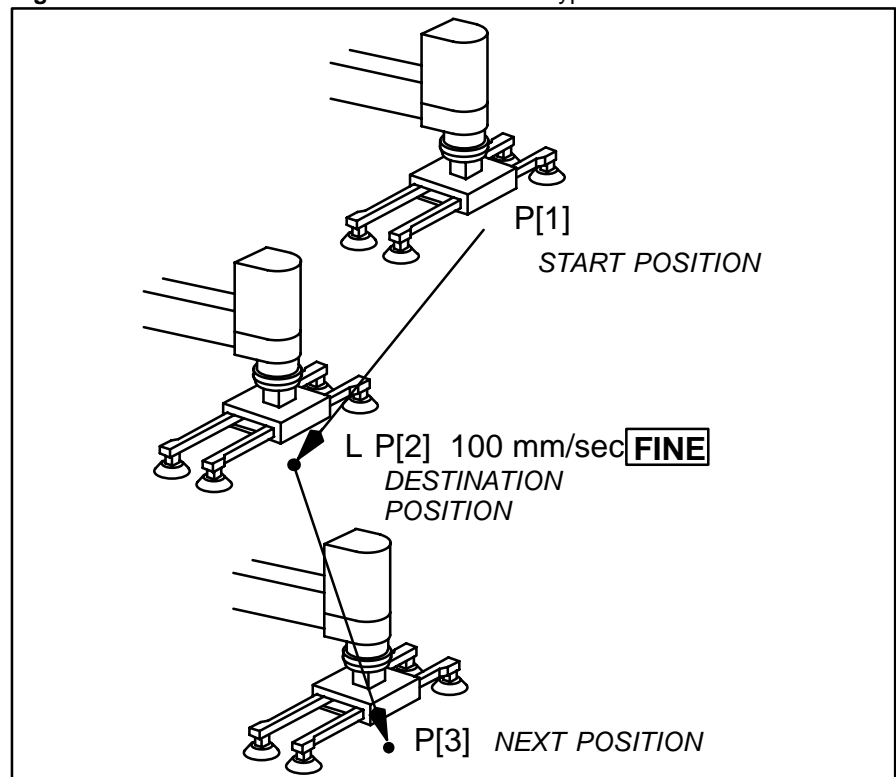
Fine Termination Type

J P[1] 50% **FINE**

Fine termination type causes the robot to **stop at the destination position** before moving to the next position.

Figure 6-15 shows how the robot will move when you specify the fine termination type.

Figure 6-15. Robot Motion with Fine Termination Type



Continuous Termination Type

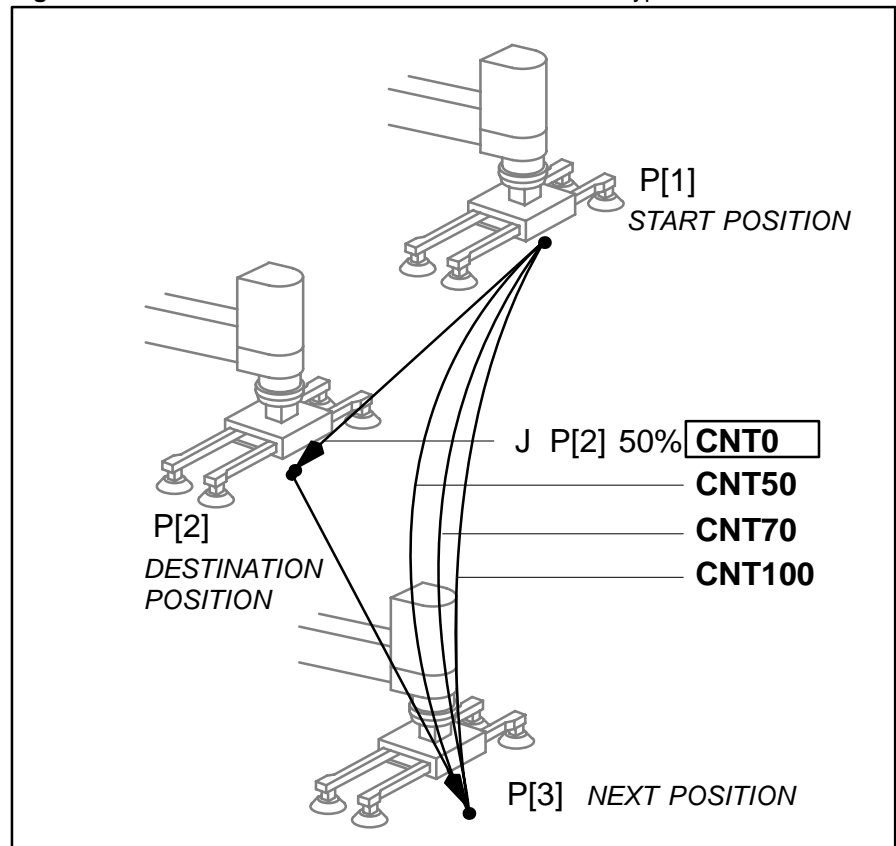
J P[1] 50% **CNT50**

Continuous termination type allows the robot to **decelerate as it approaches the destination position but does not stop** at it before it accelerates toward the next position. A value from 0 to 100 defines how close the robot comes to the destination position. At CNT0 the robot is closest, with maximum deceleration. At CNT100 the robot is farthest, with minimum deceleration.

NOTE Programming certain instructions, such as WAIT, causes the robot to stop at the destination position and execute the instruction before executing the next instruction.

Figure 6-16 shows how the robot will move with different continuous termination type values.

Figure 6-16. Robot Motion with Continuous Termination Type



6.3.6 Motion Options

Motion options can be used to provide additional information to perform specific tasks during robot motion. Motion options include

- Wrist joint motion
- Acceleration override
- Skip label
- Offset
- Offset position register
- Tool_offset
- Tool_offset position register
- Incremental motion
- EV (Extended velocity)
- Extended EV
- Simultaneous EV
- Independent EV
- Coord motion
- RTCP
- PTH switch
- SC – corner rounding
- Corner speed rate

Refer to Section 6.3.7 for information on the corner speed motion option.

Wrist Joint L P[1] 50% FINE **W/JNT**

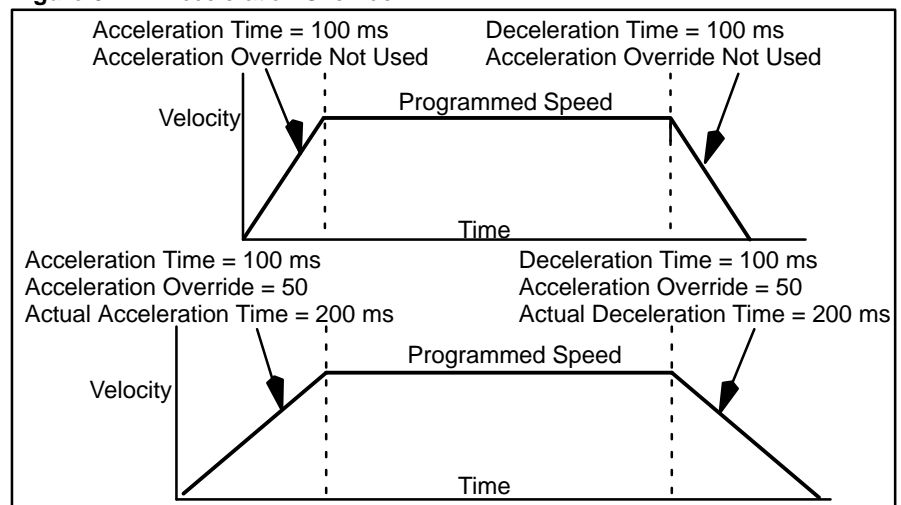
The wrist joint option is used during linear or circular moves. It causes the wrist orientation to change during moves, permitting the tool center point to move along the programmed path without flipping the wrist axes due to axis singularity points.

Acceleration OverrideJ P[1] 50% FINE **ACC50**

The acceleration override motion option specifies the acceleration/deceleration override value for each axis during motion. Acceleration override shortens or lengthens the acceleration time when the robot moves from a starting position to the destination position. Acceleration override is programmed at the destination position.

The acceleration override value ranges from 20 to 500%. This value is a percentage of the acceleration. For example, an acceleration override of 50 means the robot will take twice as long to accelerate or decelerate.

Figure 6-17 shows how the acceleration override is used.

Figure 6-17. Acceleration Override

Skip Label

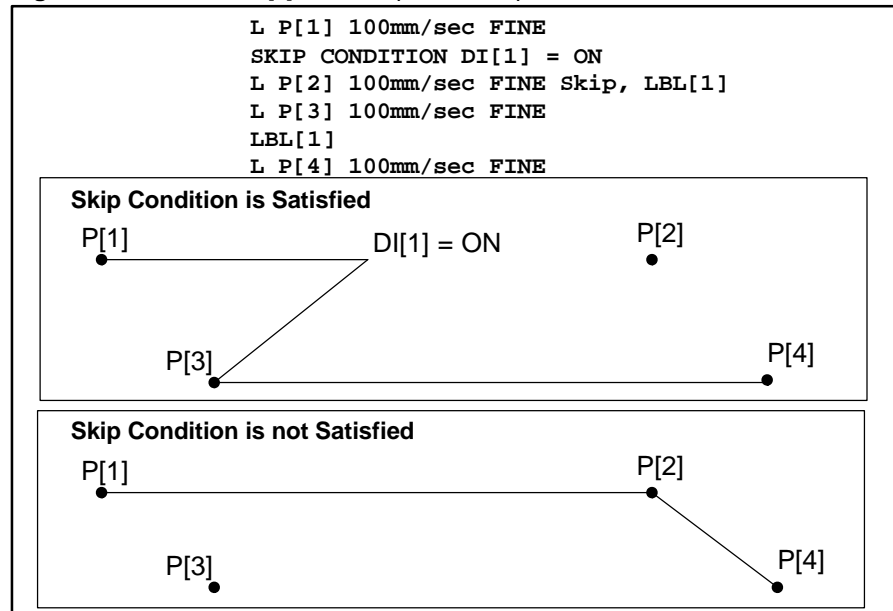
SKIP CONDITION [I/O] = [value]

J P[1] 50% FINE **Skip, LBL[3]**

The Skip, LBL[x] motion option redirects program execution based on whether a predefined SKIP CONDITION is true. A SKIP CONDITION instruction defines an example I/O condition. The execution of the motion instruction that contains the Skip, LBL[x] motion option is affected depending on status of the SKIP CONDITION, as follows:

- **If the SKIP CONDITION is satisfied**, the motion defined in the motion instruction that contains the Skip, LBL[x] motion option terminates and the next program instruction is executed.
- **If the SKIP CONDITION is not satisfied**, the motion defined in the motion instruction that contains the Skip, LBL[x] motion option is executed. After the robot reaches the destination position and the condition is still not satisfied, the program branches to the label, LBL[x].

Refer to Section 6.10 for more information on branching. Refer to Section 6.13 for more information on the SKIP CONDITION instruction. Refer to Figure 6-18 for an example of the Skip, LBL[x] motion option.

Figure 6-18. SKIP LBL[x] Motion Option Example

Offset

OFFSET CONDITION PR[x]
 J P[1] 50% FINE **Offset**

The offset motion option is used with the OFFSET CONDITION instruction to alter positional information programmed at the destination position by the offset amount specified in a position register. The OFFSET CONDITION instruction defines the position register that contains the offset information. The OFFSET CONDITION instruction must be added to the program before the offset motion instruction.

The OFFSET CONDITION instruction shown uses the offset in position register 1, PR[x]. The offset motion instruction sets the positional information to position (P[1] + PR[x]) with the orientation of P[1]. When the offset condition is set, any time the offset motion option is used, that offset will be used. Refer to Section 6.14 for more information on offset instructions.

Offset Position Register

J P[1] 50% FINE **Offset, PR[x]**

The Offset, PR[x] motion option alters positional information by the offset amount specified in the position register PR[x]. This offset affects only the motion instruction where it appears. It does not apply to any other motion instructions. The offset user frame number is the currently selected user frame number.

The OFFSET calculation depends on the position register representation specified in the OFFSET motion option:

- **If PR[x] is Cartesian representation**, the system adds each element of the position register to each element of the position to yield the position that is offset. If the position does not have Cartesian representation, the system internally converts the representation of the position to Cartesian before the offset is calculated.
- **If PR[x] is JOINT representation**, the system adds each element of the position register to each element of the position to yield the position that is offset. If the position does not have JOINT representation, the system internally converts the representation of the position to JOINT before the offset is calculated. If PR[x] is JOINT representation, an offset user frame is not used.
- **If the INC motion option is specified with the OFFSET motion option**, the position and position register **MUST** have the same representation, either Cartesian or JOINT. Before you define an offset in a motion instruction that also includes the INC motion option, make sure the representations of the position register and position are the same. For example, if the position register is JOINT representation, the position must also be JOINT representation.

Tool Offset

TOOL_OFFSET_CONDITION PR[x]
(UTOOL[1])

J P[1] 50% FINE **Tool_offset**

The Tool_offset motion option is used with the TOOL_OFFSET_CONDITION instruction to alter positional information programmed at the destination position by the tool offset amount specified in a position register. The TOOL_OFFSET_CONDITION instruction defines the position register that contains the offset information and the tool frame that will be used during the tool offset. The TOOL_OFFSET_CONDITION instruction must be added to the program before the tool offset motion instruction. Refer to Section 6.15 for more information on the TOOL_OFFSET_CONDITION instruction.

A tool offset condition instruction specifies the offset condition used in a tool offset instruction. Execute a tool offset condition instruction before you execute the corresponding tool offset instruction. After you specify the tool offset condition, it remains effective until the program terminates or the next tool offset condition instruction is executed.

When you specify tool offset conditions, be aware of the following:

- The position register specifies the direction in which the target position shifts, as well as the amount of shift.
- The tool coordinate system is used to specify offset conditions.
- When the number of a tool coordinate system is omitted, the currently selected tool coordinate system is used.
- When a motion instruction that includes a tool offset instruction is taught or a certain position is modified, the position from which the offset is subtracted can be taught.
- When a motion instruction that includes a tool offset instruction is taught or a certain position is modified, you will be asked to answer the following questions:
 - **Subtract tool offset data?**
Press YES to subtract the tool offset data from the position data and accept the new position.
Press NO to store the current position as the position data.
 - **Enter PR index of tool offset data?**
Enter the position register number specified by the tool offset condition instruction.
 - **Enter tool no. of tool offset data?**
Enter the number of the tool coordinate system in which the offset is to be specified.
- If you modify the position data manually using the numeric keys, the position is taught without subtracting the offset.
- If you teach the position from which the offset is subtracted, the current position is stored in the following cases.
 - The specified position register has not yet been initialized
 - The tool offset instruction ignore function is enabled (see other setting.)

- If you enable the ignore function for the tool offset instruction, the current position is taught as position data and you will not receive any error messages. The robot moves to the taught position, even if a tool offset instruction is executed.
- If you pause the robot during the execution of a tool offset instruction and modify the shift amount, the modified amount will be used in the resumed movement.
If you modify a position register number specified by a tool offset condition instruction, the modified number will not be used.
- In backward execution, the robot is moved to the position to which the offset has been applied. This also applies to the direct tool offset instruction, described next. Refer to Section 7.2.2, “Single Step Testing.”

Tool Offset Position Register

J P[1] 50% FINE **Tool_Offset,**
PR[2]

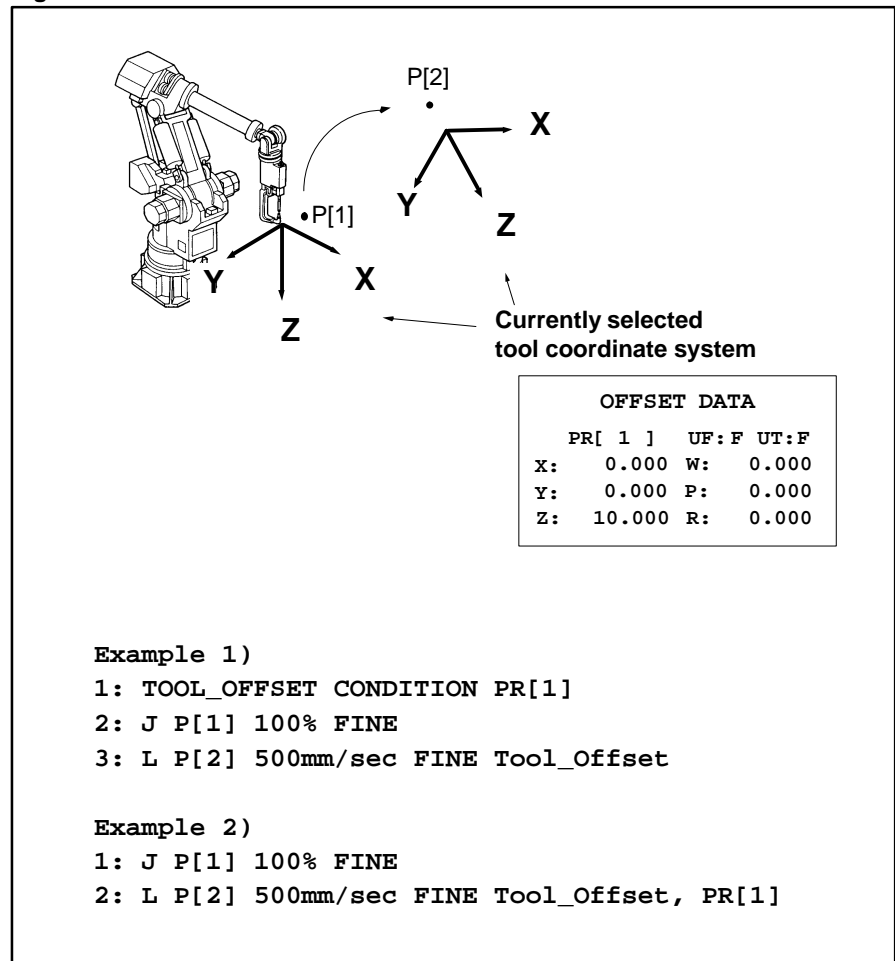
A direct tool offset instruction specifies the position register number. The robot moves according to the offset stored in the specified position register, ignoring the tool offset conditions specified by the tool offset condition instruction. The currently selected tool coordinate system is used.

When you specify tool offset position registers, be aware of the following:

- If you teach a motion instruction that includes a direct tool offset instruction or you modify a certain position, you can teach the position from which the offset is subtracted. You will be asked to answer the following question:
 - **Subtract tool offset data?**
Press YES to subtract the tool offset from the position data and accept the new position.
Press NO to store the current position as position data.
- If you modify the position data manually using the numeric keys, the position is taught without subtracting the offset.
- If you teach the position from which the offset is subtracted, the current position is stored in the following cases.
 - The specified position register has not yet been initialized
 - The direct tool offset instruction has not specified the number of a position register
 - The tool offset instruction ignore function is enabled. Refer to Section 3.16 for more information.
- If you enable the tool offset instruction ignore function, the current position is taught as position data (no prompt messages are output). The robot moves to the taught position even if a tool offset instruction is executed.

See Figure 6-19.

Figure 6-19. Tool Offset Instruction



Incremental Motion

J P[1] 50% FINE **INC**

The incremental motion option specifies that the destination position is an incremental motion amount from the previous position. See Figure 6-20 for the display of each position component.

Figure 6-20. Position Representation Screen

Position Detail				JOINT 30%	
P[1]	UF:0	UT:1	conf: N	0	0
X	29.992	mm	W -17.998	deg	
Y	.050	mm	P -10.000	deg	
Z	70.024	mm	R 0.000	deg	

2/4

Extended Velocity EV Motion Option

In addition to the programmed robot speed, the extended velocity (EV) motion option allows the specification of the programmed extended axis speed. The EV motion option has the following two options:

- Simultaneous EV
- Independent EV

Simultaneous EV J P[1] 100% Fine **EV50%**

The programmed simultaneous EV is defined as a percentage of the maximum extended axis speed (1% – 100%).

If the EV motion option is not specified, then the extended axis motion is planned based on the maximum extended axis speed. This means that the default motion without the EV option is equivalent to simultaneous motion with EV100%.

In simultaneous EV, the extended axis moves simultaneously with the robot axes. This means that they both start and end at the same time for each motion segment.

In order to achieve simultaneous motion, the robot motion time is compared with the extended axis segment time during planning. The longer time will be used for both the robot and the extended axis so that they both reach the destination at the same time.

In cases where the robot motion time is longer than the extended axis motion time, the actual extended axis speed will be lower than its programmed extended axis speed so that robot motion speed is maintained.

When the extended axis motion time is longer than the robot motion time, the actual robot speed will be slower than its programmed speed in order to maintain simultaneous motion.

When there is extended axis motion but no robot motion, the programmed extended axis speed will be used as specified, even if it could be the default maximum speed.

Independent EV J P[1] 100% Fine **Ind.EV50%**

Like simultaneous EV, the programmed independent Extended Velocity is also defined as a percentage of the maximum extended axis speed (1% – 100%).

In independent EV, the extended axis moves independently of the robot axes. Both the extended axis and the robot axes start each motion segment at the same time, however, because of their independent speed rates, they might not reach the destination at the same time. The next planned motion cannot execute until both the extended axis and the robot axes have reached the destination.

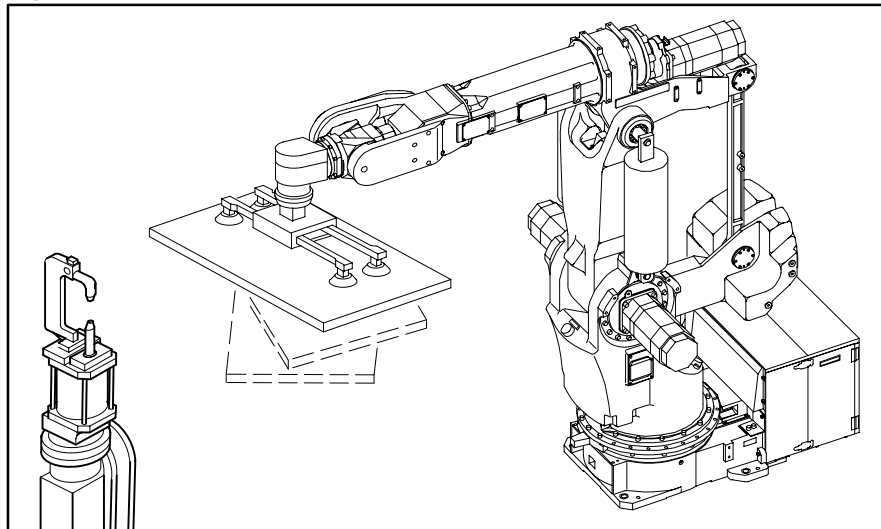
Coordinated Motion L P[1] 100mm/sec Fine **COORD**

The coordinated motion option describes motion for multiple motion groups. When this option is used, multiple motion groups move together to maintain the same position relative to each other.

Motion speed which is specified in the line is relative speed for coordinated motion. This option is effective on linear and circular motion.

Remote TCP Motion OptionL P[1] 100mm/sec CNT100 **RTCP**

The remote TCP motion option (RTCP) provides a method of controlling the orientation of the robot in applications where the tool is fixed in the workcell and the robot manipulates the workpiece around the tool. The frame used for jogging and programming is a user frame you set up and select. See Figure 6-21 for an illustration of the robot using remote TCP. The tool is fixed and the robot is holding the workpiece.

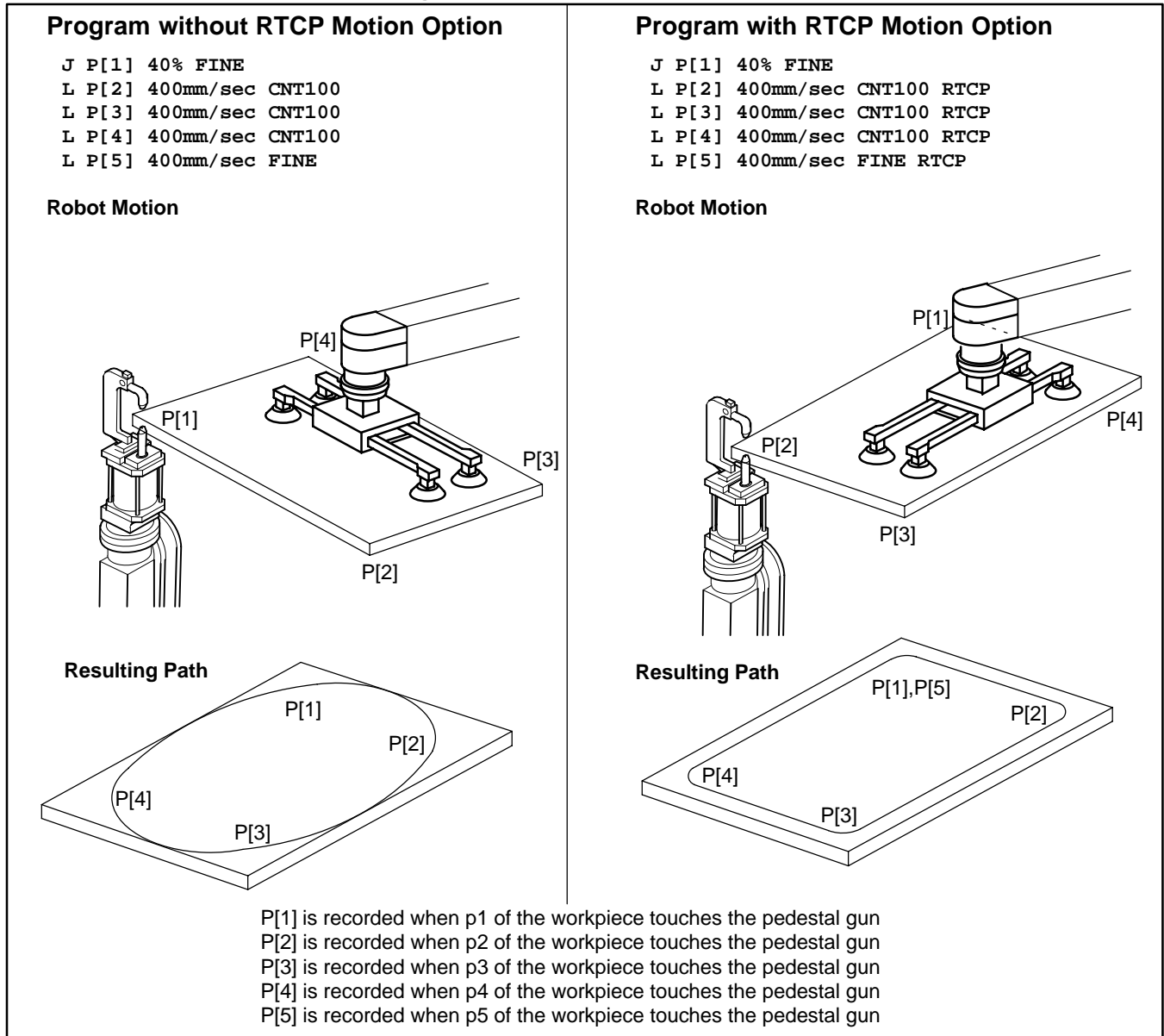
Figure 6-21. Remote TCP Motion Option

When you use remote TCP, you must first set up the user frame you will use as the remote TCP frame. When you include the remote TCP (RTCP) motion option in a motion instruction, you must specify the user frame you want to use, using the `UFRAME_NUM=` instruction, otherwise the current user frame will be used by default. Refer to Section 6.14 for more information on the `UFRAME_NUM=` instruction.

NOTE In a motion instruction that includes RTCP, the speed specified is the relative speed between the workpiece and the tool.

Figure 6-22 contains an example of how to use the RTCP motion option.

Figure 6-22. Remote TCP (RTCP) Motion Option Example



PTH Switch Motion OptionJ P[1] 50% FINE **PTH**

The PTH motion option allows you to increase the robot acceleration between positions in a series of positions, or path. You can use the PTH motion option only in motion instructions that use continuous termination type.

If you have a short series of continuous positions that are relatively close together, use the PTH motion option with each motion instruction to increase the acceleration between each position. This will reduce the amount of time the robot takes to execute that portion of the program.

The PTH motion option is effective in the following instances:

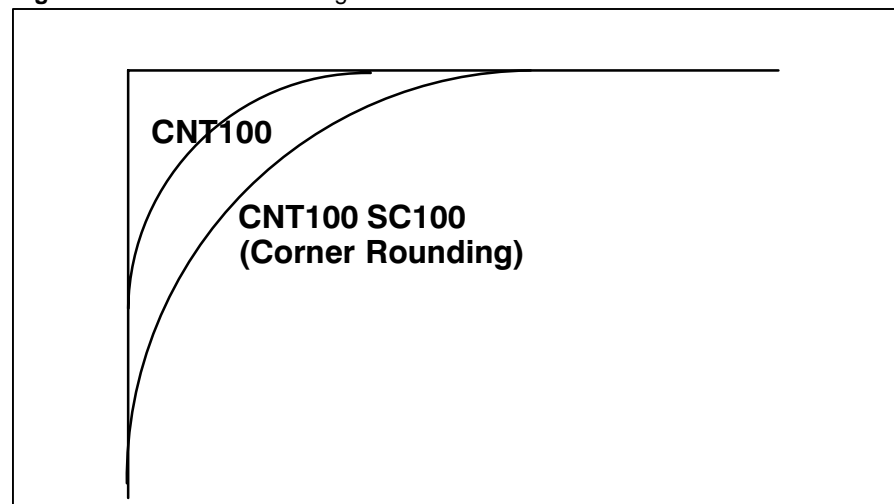
- FINE,CTN0 termtyp + Long distance motion
- CNT 1...100 termtyp + Short distance motion
- CNT 1...100 termtyp + Long distance motion

CAUTION

Some motion instructions that use the PTH switch might cause jerky motion or vibration. If the motion is attached to PTH has a vibration, delete the PTH motion option.

Corner Rounding (SC1-100)J P[1] 50% CNT100 **SC100**

The corner rounding (option) feature allows you to increase corner rounding by specifying an SC value 1-100. This feature can be used to reduce cycle time, since the resulting path is shorter. See Figure 6-23.

Figure 6-23. Corner Rounding

Corner rounding can be activated by using the SC1-100 instruction. When active the following will occur:

1. There is more corner rounding.
2. Cycle time is improved.
3. The duty cycle is improved.

The corner rounding specifies an SC value for each axis during motion. The corner rounding shortens the interval from this motion start to the next motion start. The SC value for the corner rounding is programmed at the destination position. The SC value for the corner rounding ranges from 0 to 100 (0 means no corner rounding). For example, an SC value of 100 means that the next motion starts twice as early as no SC option case.

1. The difference between ACC instruction and SC1-100 is as follows:

- ACC affects accel time only
- SC1-100 affects accel time and move time

For example,

ACC50 — accel time will be twice.

SC100 — accel time will be twice and the move time will be half.

Therefore, a value of SC100 can start the next motion earlier.

2. SC1-100 motion will be slower than no SC option case when there is only one motion instruction.

For example, for the SC100 case, accel time will be twice as for the no SC option case. Hence, for one motion instruction SC100 will be slower than no SC option motion. However, for a series of motion instructions with SC100, cycle time will be improved even though the acceleration time is twice as much as for the no SC option case. This is because the next motion instruction can start earlier, since the motion time is half of no SC option case.

(faster case)	(slower case)
10: J P[1] 100% CNT100 SC100	10: J P[1] 100% CNT100 SC100
11: J P[2] 100% CNT100 SC100	11: WAIT 1.0sec
12: J P[3] 100% CNT100 SC100	12: J P[2] 100% CNT100 SC100
:	:

3. SC option is not supported with KAREL motion.

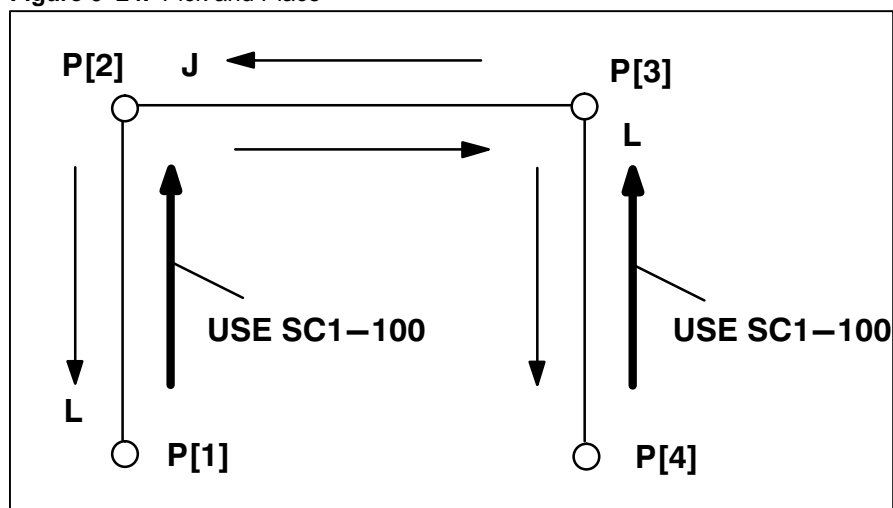
4. SC option cannot be used with AccuPath.

5. SC option can not have sufficient effect with CNT0-99 or FINE.

6. SC option is more effective with linear motion. (There are cases where using SC option with joint motion causes worse effect.)

7. SC option is recommended to be used in a pick and place as shown in Figure 6-24.

Figure 6-24. Pick and Place



The following are the guidelines to select a SC value:
Create and execute the program below.

```

10: J P[1] 100% CNT1
11: TIMER[1] = RESET
12: TIMER[1] = START
13: L P[2] 2000mm/sec CNT1
14: TIMER[1] = STOP
15: TIMER[2] = RESET
16: TIMER[2] = START
17: J P[3] 100% CNT1
18: TIMER[2] = STOP

```

If $(2 * \text{TIMER}[1]) < \text{TIMER}[2]$, then SC value for L P[2] is set to 100.
Otherwise, SC value for L P[2] is set to $(\text{TIMER}[2] * 50 / \text{TIMER}[1])$.

Limitations

Certain teach pendant instructions cause the robot to decelerate to the destination position before the next motion instruction is executed, regardless of the termination type specified. These teach pendant instructions will override the corner distance and corner speed settings. The instructions are divided into two categories:

Category 1: Instructions in this category cause the robot to decelerate, by default. However, if you override the default behavior using the LOCK PREG and UNLOCK PREG instructions, the corner path and corner speed specified will be used. The instructions in this category are as follows:

- Position register instructions: PR[], PR[] INC
- Offset instructions: OFFSET, TOOL_OFFSET

Category 2: Instructions in this category cause the robot to decelerate at all times, regardless of the termination type specified. You cannot change these default values. The instructions in this category are as follows:

- Frame instructions: UFRAME_NUM, UFRAME, UTOOL_NUM, UTOOL
- Branching instructions: IF, SELECT, CALL (if the called program uses a different motion group)
- Wait instruction: WAIT + TIMEOUT
- Miscellaneous instruction: \$PARAMETER
- Program control instructions: PAUSE, ABORT
- Macro program instruction (if the macro program uses a different motion group)
- SKIP instruction
- TRACK instruction
- Sensor instructions: RCV, SENSOR_ON, SENSOR_OFF
- Palletizing instruction: PALLETIZING-B, PL[]

6.3.7 AccuPath

AccuPath is a motion control feature that provides enhanced motion performance for **linear and circular motion** (but not joint motion) in the following areas:

- **Constant path**
With AccuPath, the robot maintains the same path regardless of static or dynamic speed override changes. A path that has been taught and tested at a low speed override will be maintained when the program is executed at 100% override.
- **Hold/Resume and Emergency Stop/Resume**
After the HOLD or EMERGENCY STOP buttons have been pressed, the robot can resume execution along the same path that was being executed prior to the HOLD or EMERGENCY STOP. The location part of the position will be along the original path, however, the orientation will be close to, but not quite exactly, along the original path.
This function requires the original path resume function to be enabled. The setting to enable original path resume function is
`$MH_ORGRSM.$RET_PTH_ENA = TRUE.`
- **Enhanced path accuracy**
The path will be executed as taught, using a straight line or circular motion.
- **Direct corner adjustment**
This allows direct corner rounding distance adjustment for each motion instruction, if you are not satisfied with the corner generated by the AccuPath motion with CNT termination type. This is provided in the corner distance termination type, CDy (where y is in mm).

- **Speed accuracy**

The robot will try to maintain the programmed speed around a corner as long as the motion is within the mechanical capability of the robot. If constant speed is not feasible, AccuPath will lower the corner speed from the programmed speed automatically. If you are not satisfied with the optimized corner speed generated by the system using the CNT termination type, you can adjust it directly using the corner speed motion option in conjunction with the corner distance termination type CDy or CNT100.

 **CAUTION**

AccuPath uses the actual payload information when calculating the corner speed. Therefore, you must set the payload correctly during installation. Otherwise, the corner speed will not operate correctly. Refer to the appropriate application-specific *Software Installation Manual* for more information.

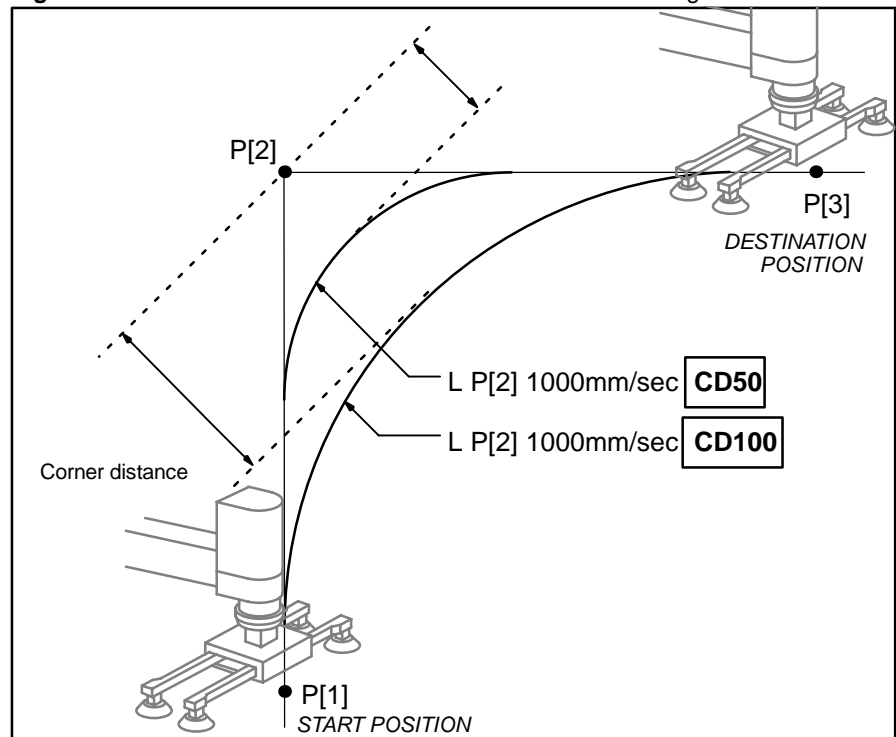
NOTE The path and speed behavior of a system using the AccuPath option are different from those systems that do not use AccuPath even if the motion instructions use the CNT termination type. Motion instructions with Fine termination type behave the same with or without AccuPath.

**Corner Distance
Termination Type,
L P[1] 100mm/sec **CDy****

If you want to adjust the corner rounding distance for a motion instruction, you can use the corner distance termination type, CDy. When you use the CD termination type, you must specify the *corner distance*.

Corner distance is the distance **from** the corner path **to** the actual taught position. See Figure 6-25.

Figure 6-25. The Effect of Corner Distance on Corner Rounding



When you set corner distance, use the following **guidelines**:

- Specify the corner distance in millimeters.
- Corner distance can range in value from 0 mm to 1000 mm.
- The **smaller** the corner distance, the **closer** the robot will get to the position, and the **less** the corner rounding.
- With a **larger** corner distance, the robot will not get as close to the position, and the **more** the corner rounding.

CAUTION

Some motion instructions that use the CDy option might cause jerky motions – especially for short distances. Occasionally, you can improve the motion by using the CSx option, adjusting the CDy parameter, or by moving the taught positions farther apart.

When you use the corner distance termination type, AccuPath will maintain constant speed if possible; otherwise, the system will slow down the robot at the corner. If you want to adjust the corner speed, use the corner speed rate motion option, described in the next section.

Corner Speed Rate Motion Option

L P[1] 100mm/sec CD100 **CSx**

By default, AccuPath will direct the robot to maintain the programmed speed around a corner, as long as this is within the mechanical capability of the robot. If constant speed is not possible, based on robot tuning, AccuPath will lower the corner speed from the programmed speed automatically. If you are not satisfied with the corner speed AccuPath provides, you can adjust the speed directly using the corner speed rate motion option, CSx.

When you set corner speed, use the following **guidelines**:

- Corner speed rate can range in value from 0% to 200%.
- A corner speed rate of **100%** is the same as the system default speed.
- A corner speed rate that is **greater than 100%** is greater than the system default speed, but less than the programmed speed.
- A corner speed rate that is **less than 100%** is less than the system default speed.

CAUTION

Some motion instructions that use the CSx motion option with a value greater than 100% might cause jerky motion or vibration. If the motion attached to CSx has a vibration, delete the CSx motion option or change the value to 100%.

Teach Pendant Instruction Limitations

Certain teach pendant instructions cause the robot to decelerate to the destination position before the next motion instruction is executed, regardless of the termination type specified. These teach pendant instructions will override the corner distance and corner speed settings. In this case, the “Fine Term Type Used” warning will be displayed. The instructions are divided into two categories:

Category 1: Instructions in this category cause the robot to decelerate, by default. However, if you override the default behavior using the LOCK PREG and UNLOCK PREG instructions, the corner path and corner speed specified will be used. The instructions in this category are as follows:

- Position register instructions: PR[], PR[] INC
- Offset instructions: OFFSET, TOOL_OFFSET

Category 2: Instructions in this category cause the robot to decelerate at all times, regardless of the termination type specified. You cannot change these default values. The instructions in this category are as follows:

- Frame instructions: UFRAME_NUM, UFRAME, UTOOL_NUM, UTOOL
- Branching instructions: IF, SELECT, CALL (if the called program uses a different motion group)
- Wait instruction: WAIT + TIMEOUT
- Miscellaneous instruction: \$PARAMETER
- Program control instructions: PAUSE, ABORT
- Macro program instruction (if the macro program uses a different motion group)
- SKIP instruction
- TRACK instruction
- Sensor instructions: RCV, SENSOR_ON, SENSOR_OFF
- Variable motion speed instructions
- Palletizing instruction: PALLETIZING-B, PL[]

Orientation Control Limitations

Orientation control limitations include

- You can only switch between default orientation control and wrist joint orientation control when FINE termtype is used. If CNT termtype is used, the previous orientation control method will be used for the current line regardless of the method specified in that line.
- If two or more taught positions are exactly the same, the robot will decelerate to the taught point regardless of the CNT value. This is consistent with the short segment half distance rule where, in this case, the half distance is 0. Refer to the “Half Distance Rule.”

AccuPath Corner

For AccuPath, a corner path is generated as follows:

- The corner path between two line segments is within the three taught positions that defines the adjacent line segments.
- For long segments, the system computes the corner path, and tries to maintain constant programmed speed around the corner path if it is within the mechanical capability of the robot (done during factory robot tuning).
- For short segments, corner path will start and end at half the distance of the shorter of the two line segments. As corner rounding reduces, constant speed around corner cannot be maintained and speed slowdown occurs.

Warning Messages

During teaching, you can check AccuPath warning messages by setting `$VC_PARAMGRP[1].$warnmessenb = TRUE`. AccuPath warning messages inform you of certain conditions of the taught path, such as “Corner speed slowdown,” and “Can’t maintain C-Dist.” These messages help you if re-teaching the path is necessary. If you set `$VC_PARAMGRP[1].$warnmessenb = TRUE`, the following error messages might be displayed:

```

MOTN-302 Corner speed slowdown
MOTN-303 Can't maintain C-Dist
MOTN-304 CD:Prog Speed achieved
MOTN-305 Can't maintain speed
MOTN-308 Can't plan corner

```

Refer to Appendix A for more details on these error messages.

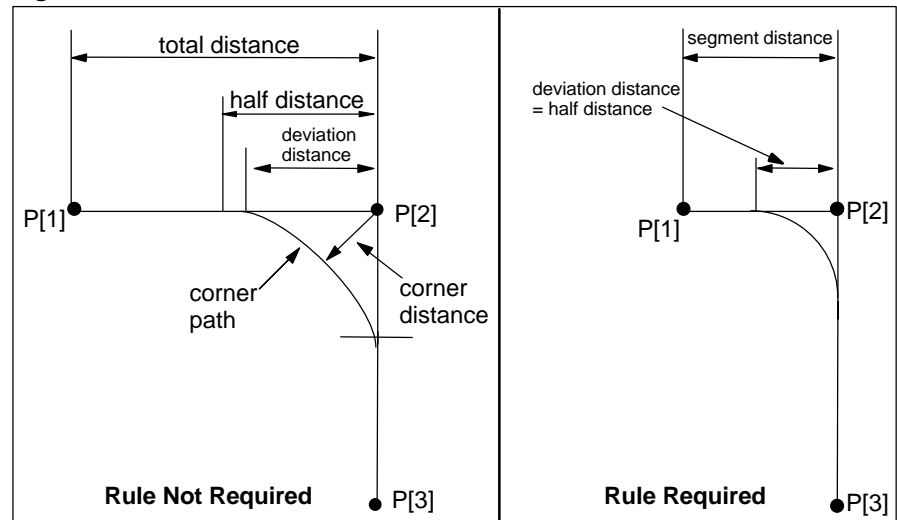
NOTE Make sure `$VC_PARAMGRP[1].$warnmessenb` is set to `FALSE` during production operation.

Half Distance Rule

As described in the “Orientation Control Limitations” section, the beginning and end of the corner path should be shorter than half the distance of the shorter of the two line segments. This is called the *half distance rule*.

In Figure 6–26, the *segment distance* refers to the distance between the taught points and the *half distance* is half of the segment distance. The *deviation distance* refers to the distance from the taught corner point P[2] to where the *corner path* deviates from the taught path. The *corner distance* is the distance from the taught corner point P[2] to the corner path.

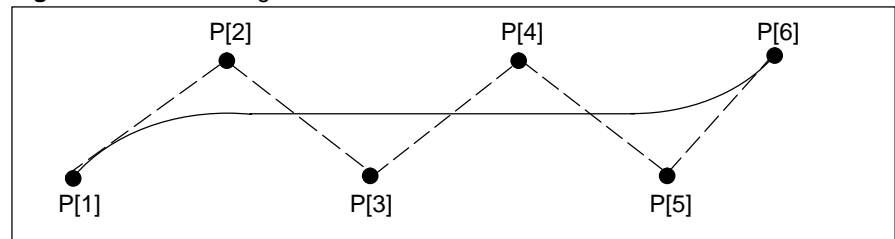
Figure 6–26. Half Distance Rule



For AccuPath, the deviation distance CAN NOT exceed the half distance. When the segment distance between taught points is short, the half distance rule is applied, in which the deviation distance is set equal to half the segment distance, as shown in Figure 6-26. As a result, the corner path is much closer to the taught point P[2], compared to the case in which the taught points are far apart.

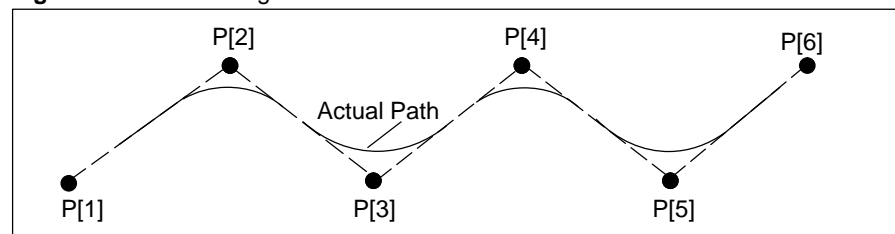
For short segments without AccuPath, as speed is increased, corner rounding is increased. Therefore, as speed is increased, the path is changed. In Figure 6-27 for example, as the speed is increased for a series of short segments, the resultant path is rounded more until, at sufficiently high speed, the path becomes a straight line in the middle segments.

Figure 6-27. Short Segment Path WITHOUT AccuPath



When AccuPath is used on a short segment, the half distance rule is applied where the corner starts and ends at a distance that is the shorter of the half segment distances that form the corner. Figure 6-28 shows the resultant path using AccuPath.

Figure 6-28. Short Segment Path with AccuPath

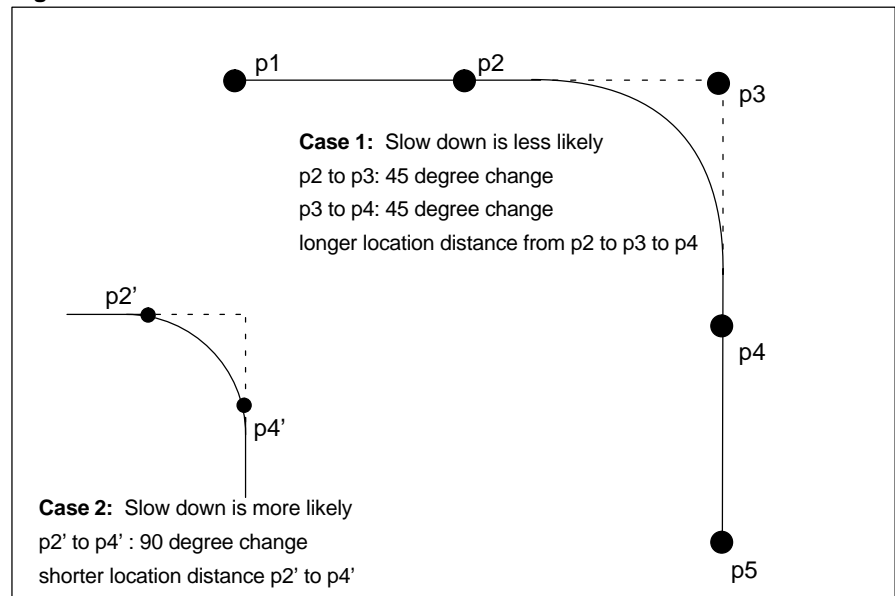


Path Orientation Guidelines

- Given two taught positions, the segment time is computed as the larger of location time and orientation time. Location time is the time to move from the start location to the destination location based on program speed. Orientation time is the time to move from start orientation to the destination orientation based on the maximum Cartesian rotation speed $\$PARAM_GROUP[.]\$rotspeedlim$.
- If orientation time is greater than location time, the effective location speed will be slower than the program speed. This is true with or without AccuPath.
- In order to achieve constant program speed around a corner with AccuPath, the orientation time must be less than the location time.
- For example, to maintain a normal approach vector with respect to the path. The objective is to make sure that the orientation time is less than the location time.

See Figure 6-29.

Figure 6-29. Path Orientation



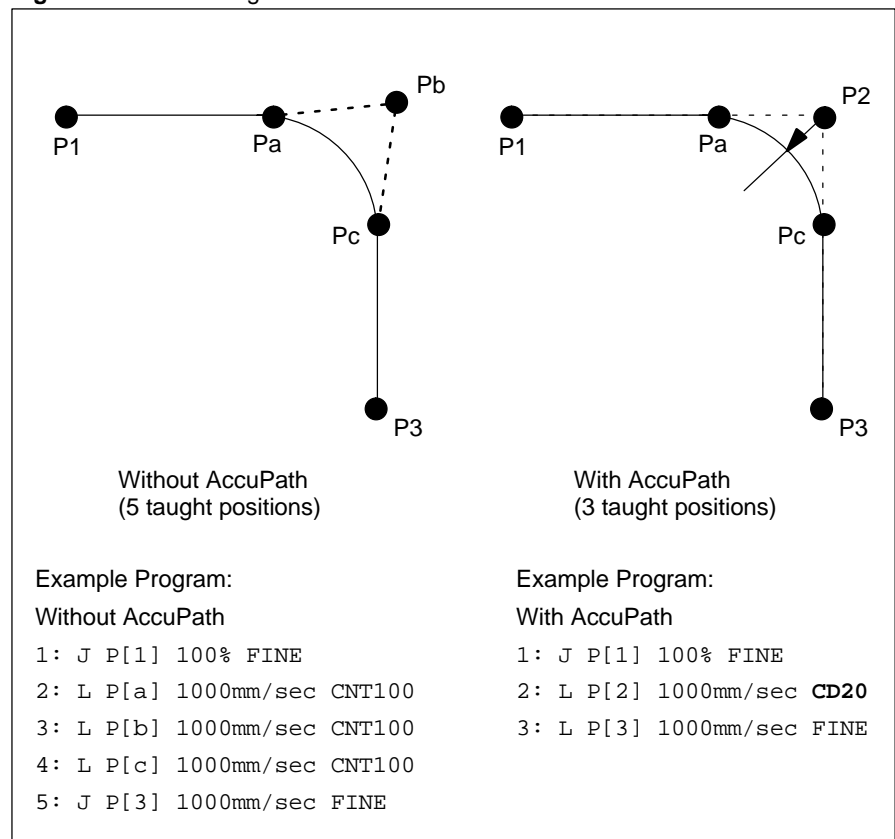
Teaching Techniques

You must be careful about the half distance rule. Keep in mind that because of the half distance rule, the specified corner distance can not be satisfied when the distance is short. The following are tips on teaching a path:

- Minimize the number of taught positions.
- Reteach positions using the CD termtyp to fit the path instead of adding positions.

Without AccuPath, you have to teach additional positions to get a small corner with high speed. Also, you have to touch up each point individually to correct any problems. With AccuPath, you do not need to do this. See Figure 6-30.

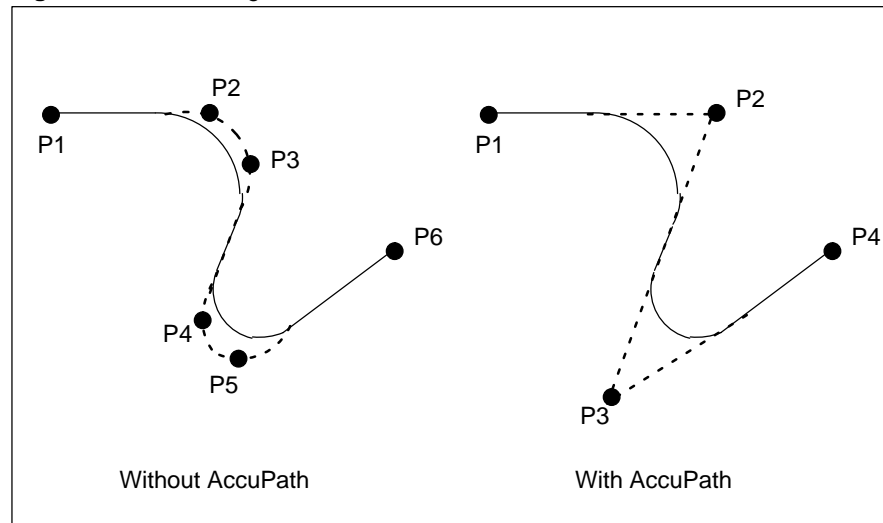
Figure 6-30. Teaching a Small Corner



Teaching a Flexible Path

When you use AccuPath, you can teach a small corner with relatively few positions. See Figure 6-31.

Figure 6-31. Teaching a Flexible Path



To teach a flexible path, you should

1. Determine the straight line which fits the tangent of the direction change point of the path.
2. Teach positions where the tangents meet.
3. Minimize the number of taught positions because of the half distance rule.
4. Use the CD termtype to specify the corner distance.

Path Verification

AccuPath can maintain the same path (x, y, z only) regardless of the speed override. But the actual path might change because of mechanical structure or motor performance. The deviation will be minimal. Therefore, you can verify the path using a small override.

To teach the path you should

1. Teach the path.
2. Run the program with a low override (10% for example).
3. If the path is not satisfactory, reteach the point.
4. Run the program using a high override. Refer to Chapter 7.

6.3.8 Correspondence between Teach Pendant Program Motion and KAREL Program Motion

The motion control functions that are supported both in the \$GROUP system variables and in the teach pendant motion instruction use the value that is specified in the teach pendant motion instruction. Refer to Table 6-4 for the correspondence between the \$GROUP system variables used for KAREL program motion and the teach pendant motion instruction.

Table 6-4. Correspondence between \$GROUP System Variables and the Teach Pendant Motion Instruction

System Variable	Teach Pendant Motion Instruction Component
\$GROUP.\$motype	Motion type
\$GROUP.\$speed	Speed – mm/sec, cm/min, inch/min
\$GROUP.\$rotspeed	Speed – deg/sec
\$GROUP.\$seg_time	Speed – sec
\$GROUP.\$termtype	Termination type
\$GROUP.\$orient_type	Wrist joint motion option
\$GROUP.\$accel_ovrd	Acceleration override (ACC) motion option
\$GROUP.\$ext_indep	Simultaneous/Independent EV motion option
\$GROUP.\$ext_speed	Simultaneous/Independent EV motion option
\$GROUP.\$cnt_shortmo	PTH motion option

The single value of the speed field in the teach pendant motion instruction can take on the function of three system variables:

- If **translational speed (mm/sec, cm/min, inch/min)** is specified, then the rotational speed (\$GROUP.\$rotspeed) is set to \$PARAM_GROUP.\$rotspeedlim. The resulting motion is limited first by the command translational speed and second by the rotational speed limit.
- If **rotational speed (deg/sec)** is specified, then the translational speed (\$GROUP.\$speed) is set to \$PARAM_GROUP.\$speedlim. The resulting motion is limited first by the command rotational speed and second by the translational speed limit.
- If **time-based motion (sec)** is specified, then the translational speed limit uses \$PARAM_GROUP.\$speedlim (or \$PARAM_GROUP.\$jntvellim for joint motion) and \$PARAM_GROUP.\$rotspeedlim as speed limits. This is similar to how KAREL programs handle time-based motion.

Refer to the *SYSTEM R-J2 Controller Software Reference Manual* and the *KAREL Reference Manual* for more detailed information on system variables and KAREL program motion.

6.4 INTELLITRAK

The IntelliTrak function improves robot path accuracy. You do not have to create a special program for IntelliTrak.

Functions

IntelliTrak performs the following functions:

- Cartesian motion control for improving path accuracy in linear motion.
- Path accuracy has been improved in circular motion and at corners, as well as in linear motion.
- IntelliTrak can be used with built-in additional axes as well as normal additional axes.
- IntelliTrak is simultaneously enabled for multiple motion groups.
- Motion option instruction `wjnt` is also supported.
- Termination type `Cnt` is valid when the robot switches from joint motion to linear or circular motion, or from a linear or circular motion to joint motion. When motions involving short travel take place in succession, however, deceleration can occur.

Guidelines and Restrictions

Note the following when using the IntelliTrak function:

- When the following functions are used, IntelliTrak is disabled automatically:
 - Continuous turn function
 - Asynchronous additional axis speed specification
 - Line Tracking
- Switching between normal attitude control and the attitude control specified by additional motion instruction `wjnt` takes place only when the termination type is `FINE`. When the termination type is `Cnt`, the current attitude control is used, regardless of whether an additional motion instruction `wjnt` is specified next.
- Before the robot can switch from joint motion to linear or circular motion, deceleration for the previous linear or circular motion must be completed. If deceleration has not yet been completed, the robot automatically waits until deceleration is complete, before starting the next linear or circular motion. Therefore, when the robot moves a short distance as part of a joint motion performed between linear or circular motions, deceleration occurs even if the termination type is set to `Cnt`.
- In linear or circular motion, the turn number (the number of turns each axis rotates) observed when a target point is reached might differ from the taught turn number. If the termination type for the target point is `Cnt` and the next motion is a joint motion, the robot decelerates to a stop as if there were a `FINE` termination type.

-
- If a programmed speed is too high, the maximum allowable speed for a certain axis might be reached. In this case, if the system variable `$PARAM_GROUP.$mot_lim_stp` is set to FALSE (default), the warning message “SRVO-026 Motor speed limit” appears. The speed is limited to the maximum allowable speed and the robot continues operating. However, the path might vary. If the system variable `$PARAM_GROUP.$mot_lim_stp` is set to TRUE, the alarm “SRVO-039 Motor speed excess” is issued, and the robot is brought to an emergency stop.
 - If the “MOTN-017 Limit error” alarm is issued in the middle of a linear or circular motion path, the robot stops immediately (without deceleration). To prevent the robot from making a sudden stop, set the system variable `$CF_PARAMGP.$chkjntlim` to TRUE. This allows limit checking to be performed so the robot decelerates less aggressively. You should only modify the `$CF_PARAMGP.$chkjntlim` system variable when you are recording positions. When you run production, set `$CF_PARAMGP.$chkjntlim` to FALSE.
 - System variables `$PARAM_GROUP.$path_accel1` and `$PARAM_GROUP.$path_accel2` must be greater than $(\$SCR.\$itp_time * \$PARAM_GROUP.\$linear_rate)$.
 - System variables `$PARAM_GROUP.$linear_rate` and `$PARAM_GROUP.$circ_rate` must have the same value.

6.5 PALLETIZING INSTRUCTIONS

Palletizing instructions tell the robot when and how to palletize. These palletizing instructions allow you to either stack the pallet or unload the pallet. There are four types of palletizing instructions:

- PALLETIZING-B
- PALLETIZING-BX
- PALLETIZING-E
- PALLETIZING-EX

To use palletizing instructions you:

1. Select one of the palletizing instructions.
2. Use the Pallet Editor to specify how the instruction will work in your program. The Pallet Editor appears only when entering or modifying a palletizing instruction.
3. Create the stack pattern by recording certain robot positions.
4. Create the route with which the robot moves to the pallet by recording certain robot positions.

Figure 6-32 shows an example of a palletizing program.

Figure 6-32. Palletizing Program Example

```

1: R[1]=0
2: PL[1]=[1,1,1]
3: PL[2]=[1,1,1]
4: THE NEXT LINE OPENS THE GRIPPER
5: RDO[1]=ON
6: RDO[2]=OFF
7: LBL[1]
8: IF SDI[4]=ON, CALL BALLET
9: J P[5] 100% CNT100
10: R[1]=R[1]+1
11: PR[3]=PR[4]
12: IF R[1]<6, JMP LBL[10]
13: IF R[1]=7, JMP LBL[10]
14: PR[3]=PR[2]
15: LBL[10]
16: PALLETIZING-B_1
17: J PAL_1[A_1] 100% FINE Offset,PR[1]
18: J PAL_1[BTM] 30% FINE Offset,PR[1]
19: CLOSE THE GRIPPER
20: RDO[1]=OFF
21: RDO[2]=ON
22: WAIT .50 (sec)
23: J PAL_1[R_1] 30% FINE Offset,PR[1]
24: PALLETIZING-END-1
25: J P[5] 100% CNT100
26: J P[1] 100% CNT100
27: WAIT SDI[10] =OFF
28: J P[2] 100% FINE
29: L P[4] 100 mm/sec FINE
30: THE NEXT LINE OPENS THE GRIPPER
31: RDO[1]=ON
32: RDO[2]=OFF
33: WAIT .50 (sec)
34: L P[2] 100mm/sec FINE
35: WAIT SDI[11]=ON
36: J P[6] 100% CNT100
37: L P[9] 100mm/sec CNT100
38: L P[7] 100mm/sec FINE
39: THE NEXT LINE CLOSES THE GRIPPER
40: RDO[1]=OFF
41: RDO[2]=ON
42: WAIT .50 (sec)
43: L P[7] 100mm/sec FINE
44: L P[9] 100mm/sec CNT100
45: J P[[6] 100% CNT100
46: J P[[1] 100% CNT100
47: J P[[5] 100% CNT100
48: J P[[12] 100% CNT100
49: L P[[10] 100mm/sec CNT100
50: J P[[11] 100% CNT100
51: L P[13] 100mm/sec CNT50
52: PALLETIZING-B_2
53: J PAL_2[A_1] 100% FINE Offset,PR[3]
54: J PAL_2[BTM] 10% FINE Offset,PR[3]
55: THE NEXT LINE CLOSES THE GRIPPER
56: RDO[1]=OFF
57: RDO[2]=ON
58: WAIT .50(sec)
59: J PAL_2[R_1] 30% FINE Offset,PR[3]
60: PALLETIZING-END-2

61: IF R[1]<>9,JMP LBL[1]
62: R[1]=0
63: PL[1]=[1,1,1]
64: PL[2]=[1,1,1]
65: LBL[2]
66: IF SDI[4]=ON, CALL BALLET
67: J P[5] 100% CNT100
68: R[1]=R[1]+1
69: PR[3]=PR[4]
70: IF R[1]<6, JMP LBL[20]
71: IF R[1]=7, JMP LBL[20]
72: PR[3]=PR[2]
73: LBL[20]
74: PALLETIZING-B_2
75: J PAL_2[A_1] 100% FINE Offset,PR[3]
76: J PAL_2[BTM] 30% FINE Offset,PR[3]
77: CLOSE THE GRIPPER
78: RDO[1]=OFF
79: RDO[2]=ON
80: WAIT .50 (sec)
81: J PAL_2[R_1] 30% FINE Offset,PR[3]
82: PALLETIZING-END_2
83: J P[5] 100% CNT100
84: J P[1] 100% CNT100
85: WAIT SDI[10] =OFF
86: J P[2] 100% FINE
87: L P[4] 100 mm/sec FINE
88: THE NEXT INE OPENSE THE GRIPPER
89: RDO[1]=ON
90: RDO[2]=OFF
91: WAIT .50 (sec)
92: L P[2] 100mm/sec FINE
93: WAIT SDI[11]=ON
94: J P[6] 100% CNT100
95: L P[9] 100mm/sec CNT100
96: L P[7] 100mm/sec FINE
97: THE NEXT LINE CLOSES THE GRIPPER
98: RDO[1]=OFF
99: RDO[2]=ON
100: WAIT .50 (sec)
101: L P[8] 100mm/sec FINE
102: L P[9] 100mm/sec
103: J P[6] 100% CNT100
104: J P[1] 100% CNT100
105: J P[5] 100% CNT100
106: J P[12] 100% CNT100
107: L P[10] 100mm/sec CNT100
108: J P[11] 100% CNT100
109: L P[13] 100mm/sec CNT50
110: PALLETIZING-B_1
111: J PAL_1[A_1] 100% FINE Offset,PR[1]
112: J PAL_1[BTM] 30% FINE Offset,PR[1]
113: THE NEXT LINE OPENS THE GRIPPER
114: RDO[1]=ON
115: RDO[2]=OFF
116: WAIT .50(sec)
117: J PAL_1[R_1] 30% FINE Offset,PR[1]
118: PALLETIZING-END_1
119: IF R[1]<>9,JMP LBL[2]
120: R[1]=0
121: JMP LBL[1]

```

6.5.1 PALLETIZING-B Instruction

The PALLETIZING-B instruction is the basic type of palletizing instruction. You use this instruction when:

- The approach and retreat routes of your robot always have the same direction and orientation. The route is the number of robot positions you record to move your robot to the pallet and then away from the pallet. See Figure 6-33.
- Your stacking pattern consists of a line pattern. A line pattern has columns, rows, and layers placed vertically, horizontally, and perpendicularly. For the line pattern, you record three end positions that indicate where the rows, columns, and layers are located. In Figure 6-33, [1,1,1] is the last route position recorded, [1,1,5], [1,3,1] and [4,1,1] are the three recorded end positions.

Figure 6-33. Example of PALLETIZING-B Instruction

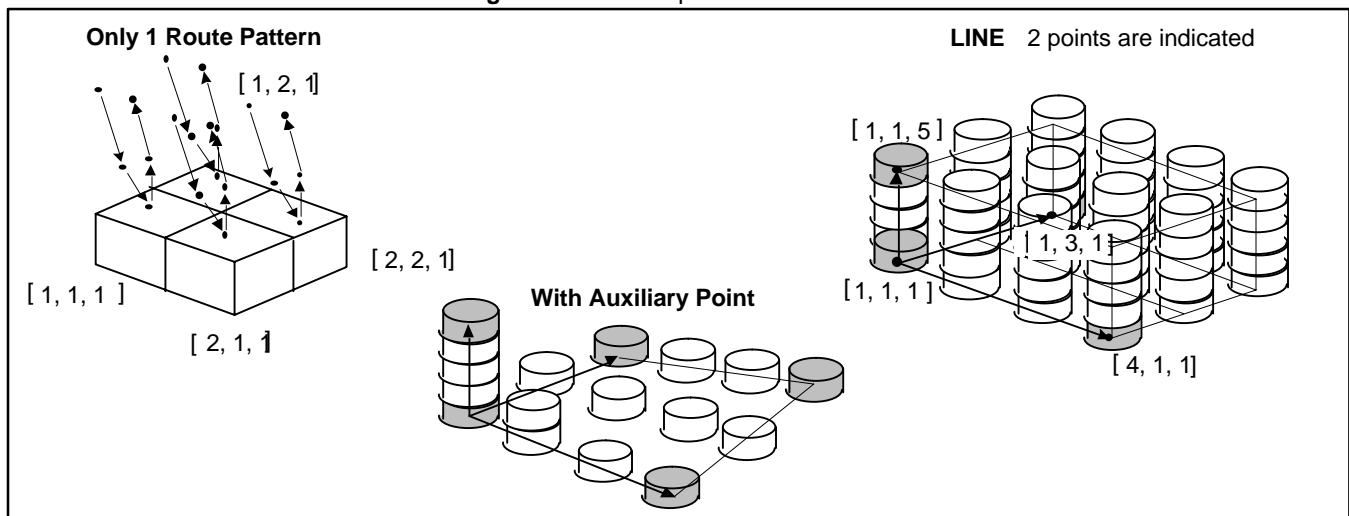


Table 6-5 lists and describes each PALLETIZING-B instruction item you can specify in the Pallet Editor.

Table 6-5. PALLETIZING-B Pallet Editor Items

ITEM	DESCRIPTION
Comment	Allows you to enter a comment about this palletizing instruction.
Type	Allows you to specify whether this palletizing instruction will palletize or depalletize. If set to PALLET , the robot stacks the pallet from bottom to top. If set to DEPALLET , the robot unloads the pallet from top to bottom.

Table 6-5. (Cont'd) PALLETIZING-B Pallet Editor Items

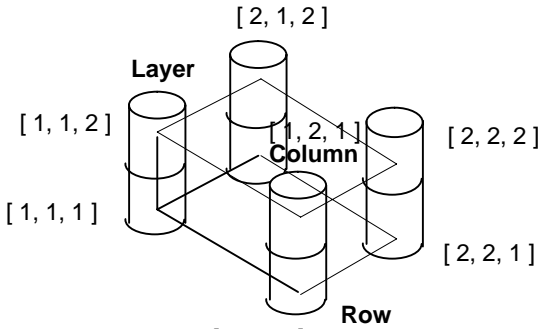
ITEM	DESCRIPTION																																																																						
<p>Incr</p>	<p>Allows you to specify how the <i>pallet register</i> will increment or decrement. This value, along with the specified <i>order</i>, controls which position will be processed by the robot next. The value of the pallet register indicates the next position to be processed. The order indicates which portion of the pallet register is updated first, second, and last.</p> <p>In the example, the specified order is row, column, layer. For palletizing, when Incr is set to 1, the instruction initializes the pallet register as the first row, first column, first layer. When Incr is set to -1 the instruction initializes the pallet register as the last row, last column, and first layer.</p> <p>For depalletizing, when Incr is set to 1, the instruction initializes the pallet register as the last row, last column, last layer. When Incr is set to -1 the instruction initializes the pallet register as the first row, first column, and last layer.</p>  <table border="1" data-bbox="581 1081 1372 1591"> <thead> <tr> <th colspan="2">TYPE = PALLET</th> <th colspan="2">TYPE = DEPALLET</th> </tr> <tr> <th></th> <th>INCR = 1</th> <th>INCR = -1</th> <th></th> <th>INCR = 1</th> <th>INCR = -1</th> </tr> <tr> <th></th> <th colspan="2">Order = RCL</th> <th></th> <th colspan="2">Order = RCL</th> </tr> </thead> <tbody> <tr> <td>Initial</td> <td>[1, 1, 1]</td> <td>[2, 2, 1]</td> <td>Initial</td> <td>[2, 2, 2]</td> <td>[1, 1, 2]</td> </tr> <tr> <td>↓</td> <td>[2, 1, 1]</td> <td>[1, 2, 1]</td> <td>↓</td> <td>[1, 2, 2]</td> <td>[2, 1, 2]</td> </tr> <tr> <td></td> <td>[1, 2, 1]</td> <td>[2, 1, 1]</td> <td></td> <td>[2, 1, 2]</td> <td>[1, 2, 2]</td> </tr> <tr> <td></td> <td>[2, 2, 1]</td> <td>[1, 1, 1]</td> <td></td> <td>[1, 1, 2]</td> <td>[2, 2, 2]</td> </tr> <tr> <td></td> <td>[1, 1, 2]</td> <td>[2, 2, 2]</td> <td></td> <td>[2, 2, 1]</td> <td>[1, 1, 2]</td> </tr> <tr> <td></td> <td>[2, 1, 2]</td> <td>[1, 2, 1]</td> <td></td> <td>[1, 2, 1]</td> <td>[2, 1, 1]</td> </tr> <tr> <td></td> <td>[1, 2, 2]</td> <td>[2, 1, 1]</td> <td></td> <td>[2, 1, 1]</td> <td>[1, 2, 1]</td> </tr> <tr> <td></td> <td>[2, 2, 2]</td> <td>[1, 1, 2]</td> <td></td> <td>[1, 1, 1]</td> <td>[2, 2, 1]</td> </tr> <tr> <td></td> <td>[1, 1, 1]</td> <td>[2, 2, 1]</td> <td></td> <td>[2, 2, 2]</td> <td>[1, 1, 2]</td> </tr> </tbody> </table>	TYPE = PALLET		TYPE = DEPALLET			INCR = 1	INCR = -1		INCR = 1	INCR = -1		Order = RCL			Order = RCL		Initial	[1, 1, 1]	[2, 2, 1]	Initial	[2, 2, 2]	[1, 1, 2]	↓	[2, 1, 1]	[1, 2, 1]	↓	[1, 2, 2]	[2, 1, 2]		[1, 2, 1]	[2, 1, 1]		[2, 1, 2]	[1, 2, 2]		[2, 2, 1]	[1, 1, 1]		[1, 1, 2]	[2, 2, 2]		[1, 1, 2]	[2, 2, 2]		[2, 2, 1]	[1, 1, 2]		[2, 1, 2]	[1, 2, 1]		[1, 2, 1]	[2, 1, 1]		[1, 2, 2]	[2, 1, 1]		[2, 1, 1]	[1, 2, 1]		[2, 2, 2]	[1, 1, 2]		[1, 1, 1]	[2, 2, 1]		[1, 1, 1]	[2, 2, 1]		[2, 2, 2]	[1, 1, 2]
TYPE = PALLET		TYPE = DEPALLET																																																																					
	INCR = 1	INCR = -1		INCR = 1	INCR = -1																																																																		
	Order = RCL			Order = RCL																																																																			
Initial	[1, 1, 1]	[2, 2, 1]	Initial	[2, 2, 2]	[1, 1, 2]																																																																		
↓	[2, 1, 1]	[1, 2, 1]	↓	[1, 2, 2]	[2, 1, 2]																																																																		
	[1, 2, 1]	[2, 1, 1]		[2, 1, 2]	[1, 2, 2]																																																																		
	[2, 2, 1]	[1, 1, 1]		[1, 1, 2]	[2, 2, 2]																																																																		
	[1, 1, 2]	[2, 2, 2]		[2, 2, 1]	[1, 1, 2]																																																																		
	[2, 1, 2]	[1, 2, 1]		[1, 2, 1]	[2, 1, 1]																																																																		
	[1, 2, 2]	[2, 1, 1]		[2, 1, 1]	[1, 2, 1]																																																																		
	[2, 2, 2]	[1, 1, 2]		[1, 1, 1]	[2, 2, 1]																																																																		
	[1, 1, 1]	[2, 2, 1]		[2, 2, 2]	[1, 1, 2]																																																																		
<p>Pal Reg</p>	<p>Allows you to specify the pallet register that will be used to store the next position to be processed.</p>																																																																						
<p>Order</p>	<p>Allows you to specify the order in which the robot will palletize or depalletize. The available options are CRL, CLR, RCL, or RLC where</p> <ul style="list-style-type: none"> • C = Column • R = Row • L = Layer 																																																																						
<p>Columns</p>	<p>Allows you to specify the total number of columns in the stack.</p>																																																																						

Table 6-5. (Cont'd) PALLETIZING-B Pallet Editor Items

ITEM	DESCRIPTION
Rows	Allows you to specify the total number of rows in the stack.
Layers	Allows you to specify the total number of layers in the stack.
Auxiliary Pos	Allows you to specify whether or not an auxiliary position will be used.
APPR	Allows you to specify how many approach route positions to use. The number you specify will be the number of positions created by the instruction for you to record.
RTRT	Allows you to specify how many retreat route positions to use in the palletizing instruction. The number you specify will be the number of positions created by the instruction for you to record.

Procedure 6-4 Using the Palletizing Editor To Enter the Palletizing-B Instruction

- Condition**
- You have created a program.
 - You are editing a program.
- Select the Instruction**
- 1 Press F1, [INST].
 - 2 Select Palletizing. See the following screen for an example.

```

PALLETIZING statem                                JOINT 10%
1 PALLETIZING-B          5 PALLETIZING-END
2 PALLETIZING-BX        6
3 PALLETIZING-E        7
4 PALLETIZING-EX       8

Main Pallet

[END]
Select item

CHOICE

```

- 3 Select PALLETIZING-B. The Pallet Editor will be displayed. See following screen for an example.

```

Main Pallet                                JOINT 10%

PALLETIZING CONFIGURATION

PALETIZING_1 [ ]
TYPE = [PALLET]          INCR = [ 1]
PAL REG = [ 1]          ORDER = [CRL]
COLUMN = [ 1]
ROWS = [ 1]
LAYERS = [ 1]
AUXILIARY POS = [NO]
APPR = [ 1] RTRT = [ 1]

PROG                                DONE

```


Set Up the Palletizing Configuration

```
Quit pallet editor?
      YES      NO
      F4      F5
```

NOTE The term Paletizing_1 in the example screen indicates that this is the first palletizing instruction in the program.

1 To return to your program at any time, press F1, PROG.

- a** Press F4, YES, to return to your program.
- b** Press F5, NO, to remain in the Pallet Editor.

2 Enter a Comment.

- a** Press ENTER.
- b** Move the cursor to select a method of naming the program: Words, Alphabetic, or Katakana.

For example, if you chose Alphabetic, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire comment is displayed.

To delete a character, press BACK SPACE.

- c** When you are finished, press ENTER.

3 Select the TYPE.

- a To palletize**, press F2, PALLET.
- b To depalletize**, press F3, DEPALL.

4 Enter the increment value.

5 Enter the number of the pallet register you want to use.

6 Enter the order. Available orders include: CRL, CLR, RCL, RLC.

- a To select rows**, press F2, R.
- b To select columns**, press F3, C.
- c To select layers**, press F4, L.

7 Enter the total number of columns.

8 Enter the total number of rows.

9 Enter the total number of layers.

10 Select the auxiliary position.

- a** To use an auxiliary position, press F2, YES.
- b** To not use an auxiliary position, press F3, NO.

11 Enter the number of approach positions.

12 Enter the number of retreat positions.

```
PALLET  DEPALL
  F2    F3
```

- 13** When you are finished setting up the PALLETIZING B instruction, press F5, DONE. See the following screen for an example.

```

Main Pallet                               JOINT 10%
TEACH BOTTOM POINTS
1: *P[1,1,1]
2: *P[10,1,1]
3: *P[1,2,1]
4: *P[1,1,2]

BACK                RECORD    DONE    >

```

NOTE The number of bottom positions you record depends on the number of columns, rows, and layers you entered in the Pallet Editor. In the above screen example, 10 columns, 2 rows, and 2 layers were entered. The order was entered as CRL.

Record Pallet Positions

```

Quit pallet editor?
YES                NO
  F4                F5

```

- 1** To return to your program at any time, press >, and then press F1, PROG.
 - a** Press F4, YES, to return to your program.
 - b** Press F5, NO, to remain in the Pallet Editor.
- 2** To return to the Pallet Editor at any time, press F1, BACK.
- 3** To record each position:
 - a** Jog the robot to the position.
 - b** Hold down the SHIFT key and press F4, RECORD.
- 4** When you are finished recording positions, press F5, DONE. See the following screen for an example.

```

Main Pallet                               JOINT 10%
PALLETIZING ROUTE POINTS
      IF PL[1] = [*,*,*}
1: J PAL_1[A_2] 30% FINE
2: J PAL_1[A_1] 30% FINE
3: J PAL_1[BTM] 30% FINE
4: J PAL_1[R_1] 30% FINE
5: J PAL_1[R_2] 30% FINE

BACK    POINT        RECORD    DONE    >

```

NOTE The number of route positions you record depends on the number of approach and retreat positions you entered in the Pallet Editor. In the above example, 2 approach and 2 retreat positions were entered.

Record Route Positions

```
Quit pallet editor?
  YES             NO
  F4             F5
```

- 1 To return to your program at any time**, press >, and then press F1, PROG.
 - a** Press F4, YES, to return to your program.
 - b** Press F5, NO, to remain in the Pallet Editor.
- 2 To return to the Pallet Editor at any time**, press F1, BACK.
- 3 To set up position default information**, press F2, POINT.
- 4 To record each position:**
 - a** Jog the robot to the position.
 - b** Hold down the SHIFT key and press F4, RECORD.
- 5** When you are finished recording positions, press F5, DONE. You are returned to your program and the PALLETIZING B instruction has been inserted.

6.5.2 PALLETIZING-BX Instruction

The PALLETIZING-BX instruction is a basic type of palletizing instruction that allows you to specify multiple route patterns. You use this instruction when:

- The approach and retreat routes of your robot have different directions and orientations. The route is the number of robot positions you record to move your robot to the pallet and then away from the pallet. See Figure 6-34.
- Your stacking pattern consists of a line pattern. A line pattern has columns, rows, and layers placed vertically, horizontally, and perpendicularly. For the line pattern, you record three end positions that indicate where the rows, columns, and layers are located. In Figure 6-34, [1,1,1] is the last route position recorded, [1,1,5], [1,3,1] and [4,1,1] are the three recorded end positions.

Figure 6-34. Example of PALLETIZING-BX Instruction

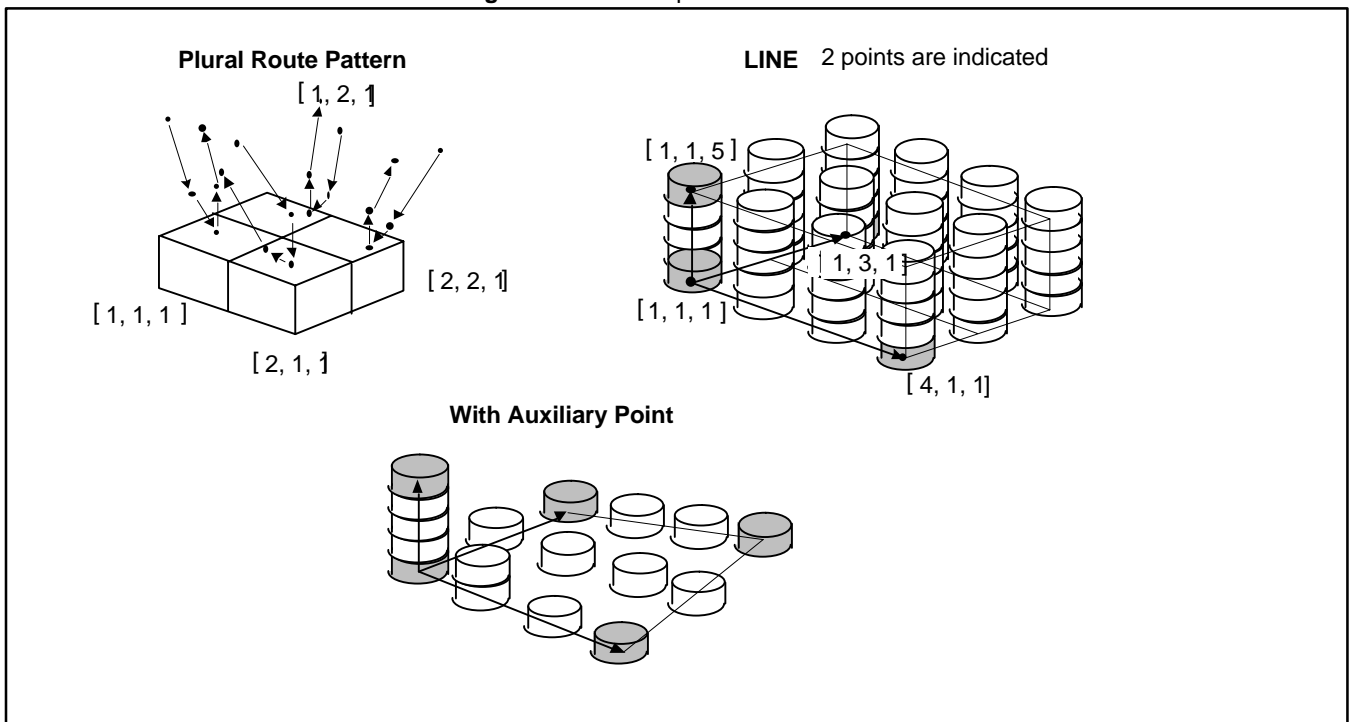


Table 6-6 lists and describes each PALLETIZING-BX instruction item you can specify in the Pallet Editor.

Table 6-6. PALLETIZING-BX Pallet Editor Items

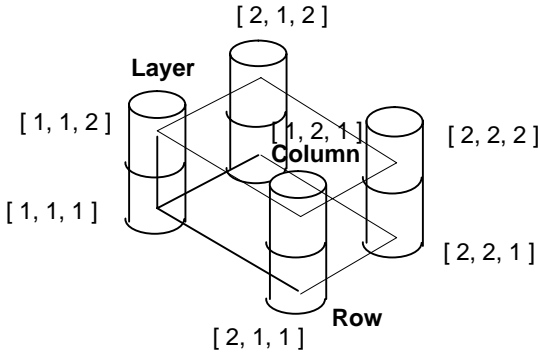
ITEM	DESCRIPTION																																																																						
Comment	Allows you to enter a comment about this palletizing instruction.																																																																						
Type	Allows you to specify whether this palletizing instruction will palletize or depalletize. If set to PALLET , the robot stacks the pallet from bottom to top. If set to DEPALLET , the robot unloads the pallet from top to bottom.																																																																						
Incr	<p>Allows you to specify how the <i>pallet register</i> will increment or decrement. This value, along with the specified <i>order</i>, controls which position will be processed by the robot next. The value of the pallet register indicates the next position to be processed. The order indicates which portion of the pallet register is updated first, second, and last.</p> <p>In the example, the specified order is row, column, layer. For palletizing, when Incr is set to 1, the instruction initializes the pallet register as the first row, first column, first layer. When Incr is set to -1 the instruction initializes the pallet register as the last row, last column, and first layer.</p> <p>For depalletizing, when Incr is set to 1, the instruction initializes the pallet register as the last row, last column, last layer. When Incr is set to -1 the instruction initializes the pallet register as the first row, first column, and last layer.</p> <div style="text-align: center;">  </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" data-bbox="581 1339 971 1365">TYPE = PALLET</th> <th colspan="2" data-bbox="987 1339 1377 1365">TYPE = DEPALLET</th> </tr> <tr> <th data-bbox="581 1365 669 1417"></th> <th data-bbox="669 1365 815 1417">INCR = 1</th> <th data-bbox="815 1365 971 1417">INCR = - 1</th> <th data-bbox="987 1365 1075 1417"></th> <th data-bbox="1075 1365 1221 1417">INCR = 1</th> <th data-bbox="1221 1365 1377 1417">INCR = - 1</th> </tr> <tr> <th data-bbox="581 1417 669 1470"></th> <th colspan="2" data-bbox="669 1417 971 1470">order = RCL</th> <th data-bbox="987 1417 1075 1470"></th> <th colspan="2" data-bbox="1075 1417 1377 1470">order = RCL</th> </tr> </thead> <tbody> <tr> <td data-bbox="581 1470 669 1522">Initial</td> <td data-bbox="669 1470 815 1522">[1, 1, 1]</td> <td data-bbox="815 1470 971 1522">[2, 2, 1]</td> <td data-bbox="987 1470 1075 1522">Initial</td> <td data-bbox="1075 1470 1221 1522">[2, 2, 2]</td> <td data-bbox="1221 1470 1377 1522">[1, 1, 2]</td> </tr> <tr> <td data-bbox="581 1522 669 1564" style="text-align: center;">↓</td> <td data-bbox="669 1522 815 1564">[2, 1, 1]</td> <td data-bbox="815 1522 971 1564">[1, 2, 1]</td> <td data-bbox="987 1522 1075 1564" style="text-align: center;">↓</td> <td data-bbox="1075 1522 1221 1564">[1, 2, 2]</td> <td data-bbox="1221 1522 1377 1564">[2, 1, 2]</td> </tr> <tr> <td data-bbox="581 1564 669 1606"></td> <td data-bbox="669 1564 815 1606">[1, 2, 1]</td> <td data-bbox="815 1564 971 1606">[2, 1, 1]</td> <td data-bbox="987 1564 1075 1606"></td> <td data-bbox="1075 1564 1221 1606">[2, 1, 2]</td> <td data-bbox="1221 1564 1377 1606">[1, 2, 2]</td> </tr> <tr> <td data-bbox="581 1606 669 1648"></td> <td data-bbox="669 1606 815 1648">[2, 2, 1]</td> <td data-bbox="815 1606 971 1648">[1, 1, 1]</td> <td data-bbox="987 1606 1075 1648"></td> <td data-bbox="1075 1606 1221 1648">[1, 1, 2]</td> <td data-bbox="1221 1606 1377 1648">[2, 2, 2]</td> </tr> <tr> <td data-bbox="581 1648 669 1690"></td> <td data-bbox="669 1648 815 1690">[1, 1, 2]</td> <td data-bbox="815 1648 971 1690">[2, 2, 2]</td> <td data-bbox="987 1648 1075 1690" style="text-align: center;">↓</td> <td data-bbox="1075 1648 1221 1690">[2, 2, 1]</td> <td data-bbox="1221 1648 1377 1690">[1, 1, 2]</td> </tr> <tr> <td data-bbox="581 1690 669 1732"></td> <td data-bbox="669 1690 815 1732">[2, 1, 2]</td> <td data-bbox="815 1690 971 1732">[1, 2, 1]</td> <td data-bbox="987 1690 1075 1732"></td> <td data-bbox="1075 1690 1221 1732">[1, 2, 1]</td> <td data-bbox="1221 1690 1377 1732">[2, 1, 1]</td> </tr> <tr> <td data-bbox="581 1732 669 1774"></td> <td data-bbox="669 1732 815 1774">[1, 2, 2]</td> <td data-bbox="815 1732 971 1774">[2, 1, 1]</td> <td data-bbox="987 1732 1075 1774"></td> <td data-bbox="1075 1732 1221 1774">[2, 1, 1]</td> <td data-bbox="1221 1732 1377 1774">[1, 2, 1]</td> </tr> <tr> <td data-bbox="581 1774 669 1816"></td> <td data-bbox="669 1774 815 1816">[2, 2, 2]</td> <td data-bbox="815 1774 971 1816">[1, 1, 2]</td> <td data-bbox="987 1774 1075 1816"></td> <td data-bbox="1075 1774 1221 1816">[1, 1, 1]</td> <td data-bbox="1221 1774 1377 1816">[2, 2, 1]</td> </tr> <tr> <td data-bbox="581 1816 669 1858"></td> <td data-bbox="669 1816 815 1858">[1, 1, 1]</td> <td data-bbox="815 1816 971 1858">[2, 2, 1]</td> <td data-bbox="987 1816 1075 1858"></td> <td data-bbox="1075 1816 1221 1858">[2, 2, 2]</td> <td data-bbox="1221 1816 1377 1858">[1, 1, 2]</td> </tr> </tbody> </table>	TYPE = PALLET		TYPE = DEPALLET			INCR = 1	INCR = - 1		INCR = 1	INCR = - 1		order = RCL			order = RCL		Initial	[1, 1, 1]	[2, 2, 1]	Initial	[2, 2, 2]	[1, 1, 2]	↓	[2, 1, 1]	[1, 2, 1]	↓	[1, 2, 2]	[2, 1, 2]		[1, 2, 1]	[2, 1, 1]		[2, 1, 2]	[1, 2, 2]		[2, 2, 1]	[1, 1, 1]		[1, 1, 2]	[2, 2, 2]		[1, 1, 2]	[2, 2, 2]	↓	[2, 2, 1]	[1, 1, 2]		[2, 1, 2]	[1, 2, 1]		[1, 2, 1]	[2, 1, 1]		[1, 2, 2]	[2, 1, 1]		[2, 1, 1]	[1, 2, 1]		[2, 2, 2]	[1, 1, 2]		[1, 1, 1]	[2, 2, 1]		[1, 1, 1]	[2, 2, 1]		[2, 2, 2]	[1, 1, 2]
TYPE = PALLET		TYPE = DEPALLET																																																																					
	INCR = 1	INCR = - 1		INCR = 1	INCR = - 1																																																																		
	order = RCL			order = RCL																																																																			
Initial	[1, 1, 1]	[2, 2, 1]	Initial	[2, 2, 2]	[1, 1, 2]																																																																		
↓	[2, 1, 1]	[1, 2, 1]	↓	[1, 2, 2]	[2, 1, 2]																																																																		
	[1, 2, 1]	[2, 1, 1]		[2, 1, 2]	[1, 2, 2]																																																																		
	[2, 2, 1]	[1, 1, 1]		[1, 1, 2]	[2, 2, 2]																																																																		
	[1, 1, 2]	[2, 2, 2]	↓	[2, 2, 1]	[1, 1, 2]																																																																		
	[2, 1, 2]	[1, 2, 1]		[1, 2, 1]	[2, 1, 1]																																																																		
	[1, 2, 2]	[2, 1, 1]		[2, 1, 1]	[1, 2, 1]																																																																		
	[2, 2, 2]	[1, 1, 2]		[1, 1, 1]	[2, 2, 1]																																																																		
	[1, 1, 1]	[2, 2, 1]		[2, 2, 2]	[1, 1, 2]																																																																		

Table 6-6. (Cont'd) PALLETIZING-BX Pallet Editor Items

ITEM	DESCRIPTION
Pal Reg	Allows you to specify the pallet register that will be used to store the next position to be processed.
Order	Allows you to specify the order in which the robot will palletize or depalletize. The available options are CRL, CLR, RCL, or RLC where <ul style="list-style-type: none"> • C = Column • R = Row • L = Layer
Columns	Allows you to specify the total number of columns in the stack.
Rows	Allows you to specify the total number of rows in the stack.
Layers	Allows you to specify the total number of layers in the stack.
Auxiliary Pos	Allows you to specify whether or not an auxiliary position will be used.
APPR	Allows you to specify how many approach route positions to use. The number you specify will be the number of positions created by the instruction for you to record.
RTRT	Allows you to specify how many retreat route positions to use in the palletizing instruction. The number you specify will be the number of positions created by the instruction for you to record.
PATTERN	Allows you to specify the number of approach and retreat routes to use. These routes patterns can be DIRECT or MODULO . A direct pattern uses the same multiple routes for each layer. A modulo pattern allows you to flip the pattern between layers.

Procedure 6-5 Using the Palletizing Editor To Enter the Palletizing-BX Instruction

- Condition**
- You have created a program.
 - You are editing a program.

- Select the Instruction**
- 1 Press F1, [INST].
 - 2 Select Palletizing. See the following screen for an example.

PALLETIZING statem		JOINT 10%
1	PALLETIZING-B	5 PALLETIZING-END
2	PALLETIZING-BX	6
3	PALLETIZING-E	7
4	PALLETIZING-EX	8
Main Pallet		
[END]		
Select item		CHOICE

- 3 Select PALLETIZING-BX. The Pallet Editor will be displayed. See following screen for an example.

```

Main Pallet                                JOINT 10%
PALLETIZING CONFIGURATION
PALETIZING_1 [REDACTED]
TYPE = [PALLET]          INCR = [ 1]
PAL REG = [ 1]          ORDER = [CRL]
COLUMN = [          ] 1]
ROWS = [          ] 1]
LAYERS = [          ] 1]
AUXILIARY POS = [NO]
APPR=[ 1] RTRT=[ 1] PATTERN=[ 1]

PROG                                DONE

```

NOTE The term Paletizing_1 in the example screen indicates that this is the first palletizing instruction in the program.

Set Up the Palletizing Configuration

```

Quit pallet editor?
YES          NO
  F4        F5

```

- 1 To return to your program at any time, press F1, PROG.

- a Press F4, YES, to return to your program.
- b Press F5, NO, to remain in the Pallet Editor.

- 2 Enter a Comment.

- a Press ENTER.
- b Move the cursor to select a method of naming the program: Words, Alphabetic, or Katakana.

For example, if you chose Alphabetic, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire program name is displayed.

To delete a character, press BACK SPACE.

- c When you are finished, press ENTER.

- 3 Select the TYPE.

- a To palletize, press F2, PALLET.
- b To depalletize, press F3, DEPALL.

- 4 Enter the increment value.

- 5 Enter the number of the pallet register you want to use.

```

PALLET  DEPALL
  F2    F3

```

- 6 Enter the order. Available orders include: CRL, CLR, RCL, RLC.
 - a To select rows, press F2, R.
 - b To select columns, press F3, C.
 - c To select layers, press F4, L.
- 7 Enter the total number of columns.
- 8 Enter the total number of rows.
- 9 Enter the total number of layers.
- 10 Select the auxiliary position.
 - a To use an auxiliary position, press F2, YES.
 - b To not use an auxiliary position, press F3, NO.
- 11 Enter the number of approach positions.
- 12 Enter the number of retreat positions.
- 13 Enter the number of approach and retreat patterns you want to use.
- 14 When you are finished setting up the PALLETIZING BX instruction, press F5, DONE. See the following screen for an example.

```
Main Pallet                                JOINT 10%
TEACH BOTTOM POINTS
1: *P[1,1,1]
2: *P[10,1,1]
3: *P[1,2,1]
4: *P[1,1,2]

BACK                                RECORD    DONE    >
```

NOTE The number of bottom positions you record depends on the number of columns, rows, and layers you entered in the Pallet Editor. In the above screen example, 10 columns, 2 rows, and 2 layers were entered. The order was entered as CRL.

Record Pallet Positions

```
Quit pallet editor?
      YES      NO
      F4      F5
```

- 1 **To return to your program at any time**, press >, and then press F1, PROG.
 - a Press F4, YES, to return to your program.
 - b Press F5, NO, to remain in the Pallet Editor.
- 2 **To return to the Pallet Editor at any time**, press F1, BACK.
- 3 **To record each position:**
 - a Jog the robot to the position.
 - b Hold down the SHIFT key and press F4, RECORD.
- 4 When you are finished recording positions, press F5, DONE. See the following screen for an example.

```
Main Pallet                                JOINT 10%
PALLETTIZING ROUTE PATTERNS
PTN [ 1] = [ *, *, * ]
PTN [ 2] = [ *, *, * ]

BACK      DIRECT      MODULO      DONE      >
```

Set Up Palletizing Route Patterns

```
Quit pallet editor?
      YES      NO
      F4      F5
```

- 1 **To return to your program at any time**, press >, and then press F1, PROG.
 - a Press F4, YES, to return to your program.
 - b Press F5, NO, to remain in the Pallet Editor.
- 2 **To return to the Pallet Editor at any time**, press F1, BACK.
- 3 **To set up each pattern,**
 - a Cursor to each asterisk.
 - b **To not alternate the patterns between layers**, press F2, DIRECT.
 - c **To alternate the patterns between layers**, press F3, MODULO.
 - d Enter the position number for each asterisk.

- 4 When you are finished recording positions, press F5, DONE. See the following screen for an example.

```

Main Pallet                                JOINT 10%
PALLETTIZING ROUTE POINTS
      IF PL[1] = [*,*,*}
1: J PAL_1[A_2] 30% FINE
2: J PAL_1[A_1] 30% FINE
3: J PAL_1[BTM] 30% FINE
4: J PAL_1[R_1] 30% FINE
5: J PAL_1[R_2] 30% FINE
BACK      POINT      RECORD      DONE >

```

NOTE The number of route positions you record depends on the number of approach and retreat positions you entered in the Pallet Editor. In the above example, 2 approach and 2 retreat positions were entered.

Record Route Positions

```

Quit pallet editor?
YES                NO
  F4              F5

```

- 1 To return to your program at any time, press >, and then press F1, PROG.
 - a Press F4, YES, to return to your program.
 - b Press F5, NO, to remain in the Pallet Editor.
- 2 To return to the Pallet Editor at any time, press F1, BACK.
- 3 To set up position default information, press F2, POINT.
- 4 To record each position:
 - a Jog the robot to the position.
 - b Hold down the SHIFT key and press F4, RECORD.
- 5 When you are finished recording positions, press F5, DONE. You are returned to your program and the PALLETIZING BX instruction has been inserted.

6.5.3 PALLETIZING-E Instruction

The PALLETIZING-E instruction is an extended type of palletizing instruction that allows you to specify different types of stacking arrangements. You use this instruction when:

- The approach and retreat route of your robot always has the same direction and orientation. The route is the number of robot positions your record to move your robot to the pallet and then away from the pallet. See Figure 6-35.
- Your stacking pattern consists of a line pattern. A line pattern has columns, rows, and layers placed vertically, horizontally, and perpendicularly. For the line pattern, you record three end positions that indicate where the rows, columns, and layers are located. In Figure 6-35, [1,1,1] is the last route position recorded, [1,1,5] [1,3,1] and [4,1,1] are the three recorded end positions.
- Your stacking pattern consists of a free pattern. A free pattern has columns, rows and layers placed vertically, horizontally and perpendicularly. For the free pattern, you record free positions where the rows, columns, and layers are located. In Figure 6-35, [1,1,1] is the last route position recorded, [1,1,5] indicates the position of the top layer, and [2,1,1], [3,1,1], and [4,1,1] indicate the position of each row.
- You want to use an auxiliary point to define the direction of the pattern.
- You want to define the orientation of the tool at the pick/place point.

Figure 6-35. Example of PALLETIZING-E Instruction

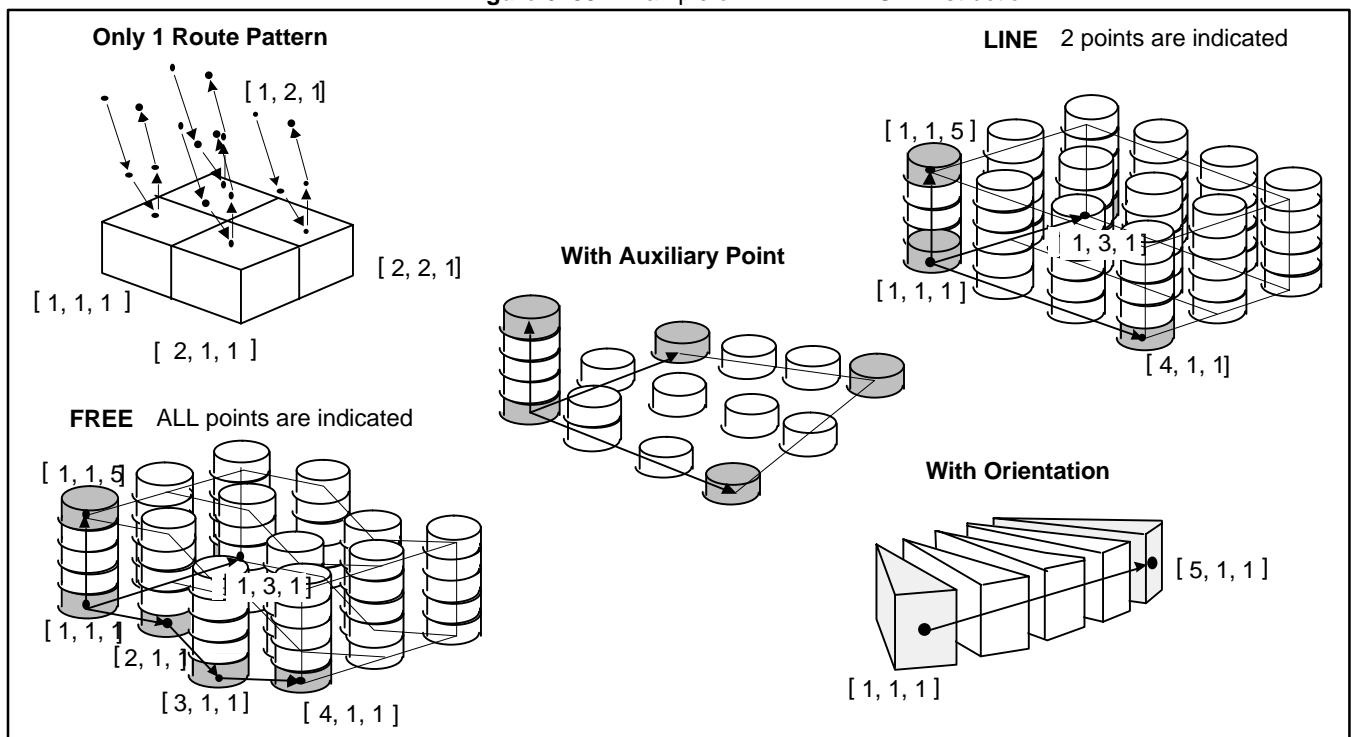


Table 6-7 lists and describes each PALLETIZING-E instruction item you can specify in the Pallet Editor.

Table 6-7. PALLETIZING-E Pallet Editor Items

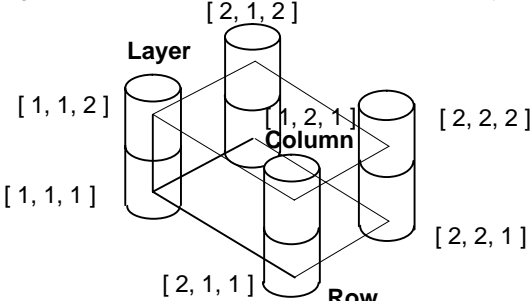
ITEM	DESCRIPTION																																																																		
Comment	Allows you to enter a comment about this palletizing instruction.																																																																		
Type	Allows you to specify whether this palletizing instruction will palletize or depalletize. If set to PALLET , the robot stacks the pallet from bottom to top. If set to DEPALLET , the robot unloads the pallet from top to bottom.																																																																		
Incr	<p>Allows you to specify how the <i>pallet register</i> will increment or decrement. This value, along with the specified <i>order</i>, controls which position will be processed by the robot next. The value of the pallet register indicates the next position to be processed. The order indicates which portion of the pallet register is updated first, second, and last.</p> <p>In the example, the specified order is row, column, layer. For palletizing, when Incr is set to 1, the instruction initializes the pallet register as the first row, first column, first layer. When Incr is set to -1 the instruction initializes the pallet register as the last row, last column, and first layer.</p> <p>For depalletizing, when Incr is set to 1, the instruction initializes the pallet register as the last row, last column, last layer. When Incr is set to -1 the instruction initializes the pallet register as the first row, first column, and last layer.</p>  <p>TYPE = PALLET TYPE = DEPALLET</p> <table border="1" data-bbox="592 1360 1372 1848"> <thead> <tr> <th></th> <th>INCR = 1</th> <th>INCR = -1</th> <th></th> <th>INCR = 1</th> <th>INCR = -1</th> </tr> <tr> <th></th> <th colspan="2">order RCL</th> <th></th> <th colspan="2">order RCL</th> </tr> </thead> <tbody> <tr> <td>Initial</td> <td>[1, 1, 1]</td> <td>[2, 2, 1]</td> <td>Initial</td> <td>[2, 2, 2]</td> <td>[1, 1, 2]</td> </tr> <tr> <td>↓</td> <td>[2, 1, 1]</td> <td>[1, 2, 1]</td> <td>↓</td> <td>[1, 2, 2]</td> <td>[2, 1, 2]</td> </tr> <tr> <td></td> <td>[1, 2, 1]</td> <td>[2, 1, 1]</td> <td></td> <td>[2, 1, 2]</td> <td>[1, 2, 2]</td> </tr> <tr> <td></td> <td>[2, 2, 1]</td> <td>[1, 1, 1]</td> <td></td> <td>[1, 1, 2]</td> <td>[2, 2, 2]</td> </tr> <tr> <td></td> <td>[1, 1, 2]</td> <td>[2, 2, 2]</td> <td></td> <td>[2, 2, 1]</td> <td>[1, 1, 2]</td> </tr> <tr> <td></td> <td>[2, 1, 2]</td> <td>[1, 2, 1]</td> <td></td> <td>[1, 2, 1]</td> <td>[2, 1, 1]</td> </tr> <tr> <td></td> <td>[1, 2, 2]</td> <td>[2, 1, 1]</td> <td></td> <td>[2, 1, 1]</td> <td>[1, 2, 1]</td> </tr> <tr> <td></td> <td>[2, 2, 2]</td> <td>[1, 1, 2]</td> <td></td> <td>[1, 1, 1]</td> <td>[2, 2, 1]</td> </tr> <tr> <td></td> <td>[1, 1, 1]</td> <td>[2, 2, 1]</td> <td></td> <td>[2, 2, 2]</td> <td>[1, 1, 2]</td> </tr> </tbody> </table>		INCR = 1	INCR = -1		INCR = 1	INCR = -1		order RCL			order RCL		Initial	[1, 1, 1]	[2, 2, 1]	Initial	[2, 2, 2]	[1, 1, 2]	↓	[2, 1, 1]	[1, 2, 1]	↓	[1, 2, 2]	[2, 1, 2]		[1, 2, 1]	[2, 1, 1]		[2, 1, 2]	[1, 2, 2]		[2, 2, 1]	[1, 1, 1]		[1, 1, 2]	[2, 2, 2]		[1, 1, 2]	[2, 2, 2]		[2, 2, 1]	[1, 1, 2]		[2, 1, 2]	[1, 2, 1]		[1, 2, 1]	[2, 1, 1]		[1, 2, 2]	[2, 1, 1]		[2, 1, 1]	[1, 2, 1]		[2, 2, 2]	[1, 1, 2]		[1, 1, 1]	[2, 2, 1]		[1, 1, 1]	[2, 2, 1]		[2, 2, 2]	[1, 1, 2]
	INCR = 1	INCR = -1		INCR = 1	INCR = -1																																																														
	order RCL			order RCL																																																															
Initial	[1, 1, 1]	[2, 2, 1]	Initial	[2, 2, 2]	[1, 1, 2]																																																														
↓	[2, 1, 1]	[1, 2, 1]	↓	[1, 2, 2]	[2, 1, 2]																																																														
	[1, 2, 1]	[2, 1, 1]		[2, 1, 2]	[1, 2, 2]																																																														
	[2, 2, 1]	[1, 1, 1]		[1, 1, 2]	[2, 2, 2]																																																														
	[1, 1, 2]	[2, 2, 2]		[2, 2, 1]	[1, 1, 2]																																																														
	[2, 1, 2]	[1, 2, 1]		[1, 2, 1]	[2, 1, 1]																																																														
	[1, 2, 2]	[2, 1, 1]		[2, 1, 1]	[1, 2, 1]																																																														
	[2, 2, 2]	[1, 1, 2]		[1, 1, 1]	[2, 2, 1]																																																														
	[1, 1, 1]	[2, 2, 1]		[2, 2, 2]	[1, 1, 2]																																																														
Pal Reg	Allows you to specify the pallet register that will be used to store the next position to be processed.																																																																		

Table 6-7. (Cont'd) PALLETIZING-E Pallet Editor Items

ITEM	DESCRIPTION
Order	Allows you to specify the order in which the robot will palletize or depalletize. The available options are CRL, CLR, RCL, or RLC where <ul style="list-style-type: none"> • C = Column • R = Row • L = Layer
Columns Rows Layers	Allows you to specify the total number of columns, rows, or layers in the stack, whether pattern is a linear or free, and whether the orientation of the TCP is a fixed or calculated orientation. <ul style="list-style-type: none"> • LINE – indicates the positions are lying in a straight line. P—P—P—P—P • FREE – indicates the positions can be anywhere in a horizontal plane. P P P P • FIX – indicates the TCP orientation equals the orientation recorded at the [1,1,1] position. • INTER – indicates the TCP orientation vector is calculated.
Auxiliary Pos	Allows you to specify whether or not an auxiliary position will be used.
APPR	Allows you to specify how many approach route positions to use. The number you specify will be the number of positions created by the instruction for you to record.
RTRT	Allows you to specify how many retreat route positions to use in the palletizing instruction. The number you specify will be the number of positions created by the instruction for you to record.

Procedure 6-6 Using the Palletizing Editor To Enter the Palletizing-E Instruction

- Condition**
- You have created a program.
 - You are editing a program.

- Select the Instruction**
- 1 Press F1, [INST].
 - 2 Select Palletizing. See the following screen for an example.

```

PALLETIZING statem                                JOINT 10%
1 PALLETIZING-B          5 PALLETIZING-END
2 PALLETIZING-BX        6
3 PALLETIZING-E         7
4 PALLETIZING-EX       8

Main Pallet
[END]
Select item                                CHOICE
    
```

- 3 Select PALLETIZING-E. The Pallet Editor will be displayed. See the following screen for an example.

```

Main Pallet                                JOINT 10%
PALLETIZING CONFIGURATION
PALETIZING_1 [REDACTED] ]
TYPE = [PALLET]          INCR = [ 1]
PAL REG = [ 1]          ORDER = [CRL]
COLUMN = [1 LINE FIX]
ROWS = [1 LINE FIX]
LAYERS = [1 LINE FIX]
AUXILIARY POS = [NO]
APPR=[ 1] RTRT=[ 1]

PROG                                DONE

```

NOTE The term Paletizing_1 in the example screen indicates that this is the first palletizing instruction in the program.

Set Up the Palletizing Configuration

```

Quit pallet editor?
YES                NO
  F4                F5

```

- 1 To return to your program at any time, press F1, PROG.

- Press F4, YES, to return to your program.
- Press F5, NO, to remain in the Pallet Editor.

- 2 Enter a Comment.

- Press ENTER.
- Move the cursor to select a method of naming the program: Words, Alphabetic, or Katakana.

For example, if you chose Alphabetic, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire program name is displayed.

To delete a character, press BACK SPACE.

- When you are finished, press ENTER.

- 3 Select the TYPE.

- To palletize,** press F2, PALLET.
- To depalletize,** press F3, DEPALL.

- 4 Enter the increment value.

- 5 Enter the number of the pallet register you want to use.

```

PALLET  DEPALL
  F2      F3

```

- 6** Enter the order. Available orders include: CRL, CLR, RCL, RLC.
 - a** To select rows, press F2, R.
 - b** To select columns, press F3, C.
 - c** To select layers, press F4, L.
- 7** Set up your column information.
 - a** Enter the total number of columns.
 - b** Select the type of positions.
 - To select linear positions, press F2, LINE.
 - To select free positions, press F3, FREE.
 - c** Select the type of TCP orientation.
 - To select fixed TCP orientation, press F2, FIX.
 - To select calculated TCP orientation, press F3, INTER.
- 8** Set up your row information.
 - a** Enter the total number of rows.
 - b** Select the type of positions.
 - To select linear positions, press F2, LINE.
 - To select free positions, press F3, FREE.
 - c** Select the type of TCP orientation.
 - To select fixed TCP orientation, press F2, FIX.
 - To select calculated TCP orientation, press F3, INTER.
- 9** Set up your layer information.
 - a** Enter the total number of layers.
 - b** Select the type of positions.
 - To select linear positions, press F2, LINE.
 - To select free positions, press F3, FREE.
 - c** Select the type of TCP orientation.
 - To select fixed TCP orientation, press F2, FIX.
 - To select calculated TCP orientation, press F3, INTER.
- 10** Select the auxiliary position.
 - a** To use an auxiliary position, press F2, YES.
 - b** To not use an auxiliary position, press F3, NO.
- 11** Enter the number of approach positions.
- 12** Enter the number of retreat positions.
- 13** Enter the number of approach and retreat patterns you want to use.

- 14 When you are finished setting up the PALLETIZING E instruction, press F5, DONE. See the following screen for an example.

```

Main Pallet                                JOINT 10%
TEACH BOTTOM POINTS
1: *P[1,1,1]
2: *P[10,1,1]
3: *P[1,2,1]
4: *P[1,1,2]

BACK                                RECORD    DONE    >

```

NOTE The number of bottom positions you record depends on the number of columns, rows, and layers you entered in the Pallet Editor. In the above screen example, 10 columns, 2 rows, and 2 layers were entered. The order was entered as CRL.

Record Pallet Positions

```

Quit pallet editor?
YES                NO
  F4                F5

```

- 1 To return to your program at any time, press >, and then press F1, PROG.
 - a Press F4, YES, to return to your program.
 - b Press F5, NO, to remain in the Pallet Editor.
- 2 To return to the Pallet Editor at any time, press F1, BACK.
- 3 To record each position:
 - a Jog the robot to the position.
 - b Hold down the SHIFT key and press F4, RECORD.
- 4 When you are finished recording positions, press F5, DONE. See the following screen for an example.

```

Main Pallet                                JOINT 10%
PALLETIZING ROUTE POINTS
      IF PL[1] = [*,*,*}
1: J PAL_1[A_2] 30% FINE
2: J PAL_1[A_1] 30% FINE
3: J PAL_1[BTM] 30% FINE
4: J PAL_1[R_1] 30% FINE
5: J PAL_1[R_2] 30% FINE

BACK    POINT    RECORD    DONE    >

```

NOTE The number of route positions you record depends on the number of approach and retreat positions you entered in the Pallet Editor. In the above example, 2 approach and 2 retreat positions were entered.

Record Route Positions

```
Quit pallet editor?
  YES              NO
  F4              F5
```

- 1 To return to your program at any time**, press >, and then press F1, PROG.
 - a** Press F4, YES, to return to your program.
 - b** Press F5, NO, to remain in the Pallet Editor.
- 2 To return to the Pallet Editor at any time**, press F1, BACK.
- 3 To set up position default information**, press F2, POINT.
- 4 To record each position:**
 - a** Jog the robot to the position.
 - b** Hold down the SHIFT key and press F4, RECORD.
- 5** When you are finished recording positions, press F5, DONE. You are returned to your program and the PALLETIZING E instruction has been inserted.

6.5.4 PALLETIZING-EX Instruction

The PALLETIZING-EX instruction can be used in the following cases:

- The approach and retreat routes of your robot have different directions and orientations. The route is the number of robot positions you record to move your robot to the pallet and then away from the pallet. See Figure 6-36.
- If your stacking pattern consists of a line pattern. A line pattern has columns, rows, and layers placed vertically, horizontally, and perpendicularly. For the line pattern, you record three end positions that indicate where the rows, columns, and layers are located. In Figure 6-36, [1,1,1] is the last route position recorded, [1,1,5] [1,3,1] and [4,1,1] are the three recorded end positions.
- If your stacking pattern consists of a free pattern. A free pattern has columns, rows and layers placed vertically, horizontally, and perpendicularly. For the free pattern, you record two end positions where the rows, columns, and layers are located. In Figure 6-36, [1,1,1] is the last route position recorded, [1,1,5] indicates the position of the top layer, and [2,1,1], [3,1,1], and [4,1,1] indicate the position of each row.
- Your layer patterns alternate. See Figure 6-36.
- You want to use an auxiliary position to define the direction of the pattern.
- You want to define the orientation of the tool at the pick/place point.

Figure 6-36. Example of PALLETIZING-EX Instruction

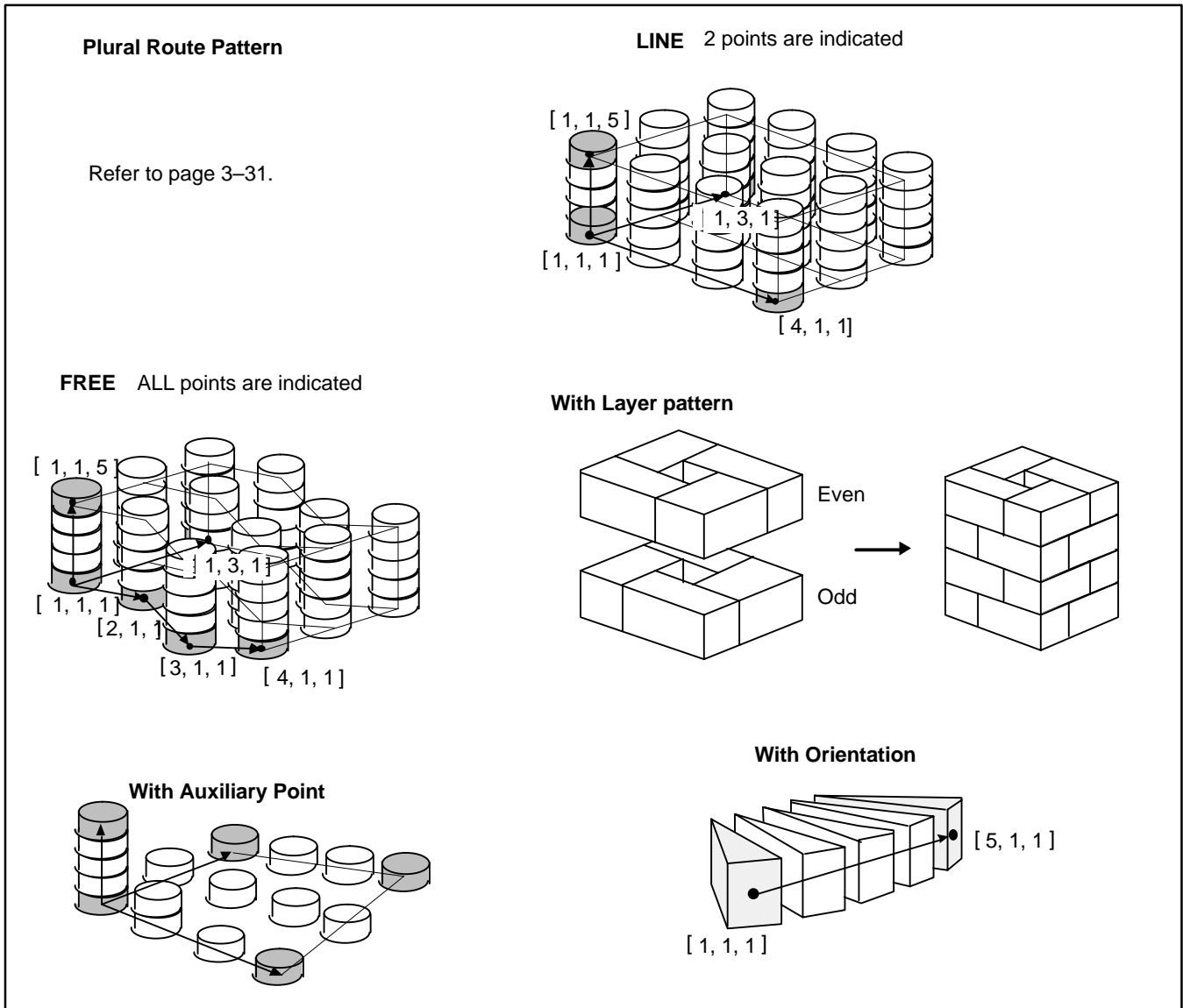


Table 6-8 lists and describes each PALLETIZING-BX instruction item you can specify in the Pallet Editor.

Table 6-8. PALLETIZING-BX Pallet Editor Items

ITEM	DESCRIPTION
Comment	Allows you to enter a comment about this palletizing instruction.
Type	Allows you to specify whether this palletizing instruction will palletize or depalletize. If set to PALLET , the robot stacks the pallet from bottom to top. If set to DEPALLET , the robot unloads the pallet from top to bottom.

Table 6-8. (Cont'd) PALLETIZING-BX Pallet Editor Items

ITEM	DESCRIPTION																																																							
<p>Incr</p>	<p>Allows you to specify how the <i>pallet register</i> will increment or decrement. This value, along with the specified <i>order</i>, controls which position will be processed by the robot next. The value of the pallet register indicates the next position to be processed. The order indicates which portion of the pallet register is updated first, second, and last.</p> <p>In the example, the specified order is row, column, layer. For palletizing, when Incr is set to 1, the instruction initializes the pallet register as the first row, first column, first layer. When Incr is set to -1 the instruction initializes the pallet register as the last row, last column, and first layer.</p> <p>For depalletizing, when Incr is set to 1, the instruction initializes the pallet register as the last row, last column, last layer. When Incr is set to -1 the instruction initializes the pallet register as the first row, first column, and last layer.</p> <div style="text-align: center;"> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th colspan="2" style="text-align: center;">TYPE = PALLET</th> <th colspan="2" style="text-align: center;">TYPE = DEPALLET</th> </tr> <tr> <th></th> <th style="text-align: center;">INCR = 1</th> <th style="text-align: center;">INCR = - 1</th> <th></th> <th style="text-align: center;">INCR = 1 INCR = - 1</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Initial</td> <td style="text-align: center;">[1, 1, 1]</td> <td style="text-align: center;">[2, 2, 1]</td> <td style="text-align: center;">Initial</td> <td style="text-align: center;">[2, 2, 2] [1, 1, 2]</td> </tr> <tr> <td style="text-align: center;">↓</td> <td style="text-align: center;">[2, 1, 1]</td> <td style="text-align: center;">[1, 2, 1]</td> <td style="text-align: center;">↓</td> <td style="text-align: center;">[1, 2, 2] [2, 1, 2]</td> </tr> <tr> <td></td> <td style="text-align: center;">[1, 2, 1]</td> <td style="text-align: center;">[2, 1, 1]</td> <td></td> <td style="text-align: center;">[2, 1, 2] [1, 2, 2]</td> </tr> <tr> <td></td> <td style="text-align: center;">[2, 2, 1]</td> <td style="text-align: center;">[1, 1, 1]</td> <td></td> <td style="text-align: center;">[1, 1, 2] [2, 2, 2]</td> </tr> <tr> <td></td> <td style="text-align: center;">[1, 1, 2]</td> <td style="text-align: center;">[2, 2, 2]</td> <td></td> <td style="text-align: center;">[2, 2, 1] [1, 1, 2]</td> </tr> <tr> <td></td> <td style="text-align: center;">[2, 1, 2]</td> <td style="text-align: center;">[1, 2, 1]</td> <td></td> <td style="text-align: center;">[1, 2, 1] [2, 1, 1]</td> </tr> <tr> <td></td> <td style="text-align: center;">[1, 2, 2]</td> <td style="text-align: center;">[2, 1, 1]</td> <td></td> <td style="text-align: center;">[2, 1, 1] [1, 2, 1]</td> </tr> <tr> <td></td> <td style="text-align: center;">[2, 2, 2]</td> <td style="text-align: center;">[1, 1, 2]</td> <td></td> <td style="text-align: center;">[1, 1, 1] [2, 2, 1]</td> </tr> <tr> <td></td> <td style="text-align: center;">[1, 1, 1]</td> <td style="text-align: center;">[2, 2, 1]</td> <td></td> <td style="text-align: center;">[2, 2, 2] [1, 1, 2]</td> </tr> </tbody> </table>		TYPE = PALLET		TYPE = DEPALLET			INCR = 1	INCR = - 1		INCR = 1 INCR = - 1	Initial	[1, 1, 1]	[2, 2, 1]	Initial	[2, 2, 2] [1, 1, 2]	↓	[2, 1, 1]	[1, 2, 1]	↓	[1, 2, 2] [2, 1, 2]		[1, 2, 1]	[2, 1, 1]		[2, 1, 2] [1, 2, 2]		[2, 2, 1]	[1, 1, 1]		[1, 1, 2] [2, 2, 2]		[1, 1, 2]	[2, 2, 2]		[2, 2, 1] [1, 1, 2]		[2, 1, 2]	[1, 2, 1]		[1, 2, 1] [2, 1, 1]		[1, 2, 2]	[2, 1, 1]		[2, 1, 1] [1, 2, 1]		[2, 2, 2]	[1, 1, 2]		[1, 1, 1] [2, 2, 1]		[1, 1, 1]	[2, 2, 1]		[2, 2, 2] [1, 1, 2]
	TYPE = PALLET		TYPE = DEPALLET																																																					
	INCR = 1	INCR = - 1		INCR = 1 INCR = - 1																																																				
Initial	[1, 1, 1]	[2, 2, 1]	Initial	[2, 2, 2] [1, 1, 2]																																																				
↓	[2, 1, 1]	[1, 2, 1]	↓	[1, 2, 2] [2, 1, 2]																																																				
	[1, 2, 1]	[2, 1, 1]		[2, 1, 2] [1, 2, 2]																																																				
	[2, 2, 1]	[1, 1, 1]		[1, 1, 2] [2, 2, 2]																																																				
	[1, 1, 2]	[2, 2, 2]		[2, 2, 1] [1, 1, 2]																																																				
	[2, 1, 2]	[1, 2, 1]		[1, 2, 1] [2, 1, 1]																																																				
	[1, 2, 2]	[2, 1, 1]		[2, 1, 1] [1, 2, 1]																																																				
	[2, 2, 2]	[1, 1, 2]		[1, 1, 1] [2, 2, 1]																																																				
	[1, 1, 1]	[2, 2, 1]		[2, 2, 2] [1, 1, 2]																																																				
<p>Pal Reg</p>	<p>Allows you to specify the pallet register that will be used to store the next position to be processed.</p>																																																							
<p>Order</p>	<p>Allows you to specify the order in which the robot will palletize or depalletize. The available options are CRL, CLR, RCL, or RLC where</p> <ul style="list-style-type: none"> • C = Column • R = Row • L = Layer 																																																							

Table 6-8. (Cont'd) PALLETIZING-BX Pallet Editor Items

ITEM	DESCRIPTION
Columns Rows Layers	<p>Allows you to specify the total number of columns, rows, or layers in the stack, whether pattern is a linear or free, and whether the orientation of the TCP is a fixed or calculated orientation.</p> <ul style="list-style-type: none"> • LINE – indicates the positions are lying in a straight line. P—P—P—P—P • FREE – indicates the positions can be anywhere in a horizontal plane. P P P P P P • FIX – indicates the TCP orientation equals the orientation recorded at the [1,1,1] position. • INTER – indicates the TCP orientation vector is calculated.
Auxiliary Pos	Auxiliary Pos cannot be specified for the PALLETIZING-BX instruction.
APPR	Allows you to specify how many approach route positions to use. The number you specify will be the number of positions created by the instruction for you to record.
RTRT	Allows you to specify how many retreat route positions to use in the palletizing instruction. The number you specify will be the number of positions created by the instruction for you to record.
PATTERN	Allows you to specify the number of approach and retreat routes to use. These routes patterns can be DIRECT or MODULO . A direct pattern uses the same multiple routes for each layer. A modulo pattern allows you to alternate the multiple route patterns between layers.

Procedure 6-7 Using the Palletizing Editor To Enter the Palletizing-EX Instruction

- Condition**
- You have created a program.
 - You are editing a program.

- Select the Instruction**
- 1 Press F1, [INST].
 - 2 Select Palletizing. See the following screen for an example.

PALLETIZING statem		JOINT 10%
1	PALLETIZING-B	5 PALLETIZING-END
2	PALLETIZING-BX	6
3	PALLETIZING-E	7
4	PALLETIZING-EX	8
Main Pallet		
[END]		
Select item		CHOICE

- 3 Select PALLETIZING-EX. The Pallet Editor will be displayed. See the following screen for an example.

```

Main Pallet                                JOINT 10%
PALLETIZING CONFIGURATION
PALETIZING_1 [REDACTED] ]
TYPE = [PALLET]          INCR = [ 1]
PAL REG = [ 1]          ORDER = [CRL]
COLUMN = [1 LINE FIX]
ROWS = [1 LINE FIX]
LAYERS = [1 LINE FIX]
AUXILIARY POS = [NO]
APPR=[ 1] RTRT=[ 1]

PROG                                DONE

```

NOTE The term Paletizing_1 in the example screen indicates that this is the first palletizing instruction in the program.

Set Up the Palletizing Configuration

```

Quit pallet editor?
YES                NO
  F4                F5

```

- 1 To return to your program at any time, press F1, PROG.

- a Press F4, YES, to return to your program.
- b Press F5, NO, to remain in the Pallet Editor.

- 2 Enter a Comment.

- a Press ENTER.
- b Move the cursor to select a method of naming the program: Words, Alphabetic, or Katakana.

For example, if you chose Alphabetic, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire comment is displayed.

To delete a character, press BACK SPACE.

- c When you are finished, press ENTER.

- 3 Select the TYPE.

- a To palletize, press F2, PALLET.
- b To depalletize, press F3, DEPALL.

- 4 Enter the increment value.

- 5 Enter the number of the pallet register you want to use.

```

PALLET  DEPALL
  F2      F3

```

- 6** Enter the order. Available orders include: CRL, CLR, RCL, RLC.
 - a** To select rows, press F2, R.
 - b** To select columns, press F3, C.
 - c** To select layers, press F4, L.
- 7** Set up your column information.
 - a** Enter the total number of columns.
 - b** Select the type of positions.
 - To select linear positions, press F2, LINE.
 - To select free positions, press F3, FREE.
 - c** Select the type of TCP orientation.
 - To select fixed TCP orientation, press F2, FIX.
 - To select calculated TCP orientation, press F3, INTER.
- 8** Set up your row information.
 - a** Enter the total number of rows.
 - b** Select the type of positions.
 - To select linear positions, press F2, LINE.
 - To select free positions, press F3, FREE.
 - c** Select the type of TCP orientation.
 - To select fixed TCP orientation, press F2, FIX.
 - To select calculated TCP orientation, press F3, INTER.
- 9** Set up your layer information.
 - a** Enter the total number of layers.
 - b** Select the type of positions.
 - To select linear positions, press F2, LINE.
 - To select free positions, press F3, FREE.
 - c** Select the type of TCP orientation.
 - To select fixed TCP orientation, press F2, FIX.
 - To select calculated TCP orientation, press F3, INTER.
- 10** Select the auxiliary position.
 - a** To use an auxiliary position, press F2, YES.
 - b** To not use an auxiliary position, press F3, NO.
- 11** Enter the number of approach positions.
- 12** Enter the number of retreat positions.
- 13** Enter the number of approach and retreat patterns you want to use.

- 14 When you are finished setting up the PALLETIZING EX instruction, press F5, DONE. See the following screen for an example.

```

Main Pallet                                JOINT 10%
TEACH BOTTOM POINTS
1: *P[1,1,1]
2: *P[10,1,1]
3: *P[1,2,1]
4: *P[1,1,2]

BACK                                RECORD    DONE    >

```

NOTE The number of bottom positions you record depends on the number of columns, rows, and layers you entered in the Pallet Editor. In the above screen example, 10 columns, 2 rows, and 2 layers were entered. The order was entered as CRL.

Record Pallet Positions

```

Quit pallet editor?
YES                NO
  F4                F5

```

- 1 To return to your program at any time, press >, and then press F1, PROG.
 - a Press F4, YES, to return to your program.
 - b Press F5, NO, to remain in the Pallet Editor.
- 2 To return to the Pallet Editor at any time, press F1, BACK.
- 3 To record each position:
 - a Jog the robot to the position.
 - b Hold down the SHIFT key and press F4, RECORD.
- 4 When you are finished recording positions, press F5, DONE. See the following screen for an example.

```

Main Pallet                                JOINT 10%
PALLETIZING ROUTE PATTERNS
PTN [ 1 ] = [ *, *, * ]
PTN [ 2 ] = [ *, *, * ]

BACK    DIRECT    MODULO    DONE    >

```


Set Up Palletizing Route Patterns

```
Quit pallet editor?
      YES      NO
      F4      F5
```

- 1 To return to your program at any time, press >, and then press F1, PROG.
 - a Press F4, YES, to return to your program.
 - b Press F5, NO, to remain in the Pallet Editor.
- 2 To return to the Pallet Editor at any time, press F1, BACK.
- 3 To set up each pattern,
 - a Cursor to each asterisk.
 - b To not alternate the patterns between layers, press F2, DIRECT.
 - c To alternate the patterns between layers, press F3, MODULO.
 - d Enter the position number for each asterisk.
- 4 When you are finished recording positions, press F5, DONE. See the following screen for an example.

```
Main Pallet                                JOINT 10%
PALLETTIZING ROUTE POINTS
      IF PL[1] = [*,*,*}
1: J PAL_1[A_2] 30% FINE
2: J PAL_1[A_1] 30% FINE
3: J PAL_1[BTM] 30% FINE
4: J PAL_1[R_1] 30% FINE
5: J PAL_1[R_2] 30% FINE
BACK      POINT      RECORD      DONE >
```

NOTE The number of route positions you record depends on the number of approach and retreat positions you entered in the Pallet Editor. In the above example, 2 approach and 2 retreat positions were entered.

Record Route Positions

```
Quit pallet editor?
      YES      NO
      F4      F5
```

- 1 To return to your program at any time, press >, and then press F1, PROG.
 - a Press F4, YES, to return to your program.
 - b Press F5, NO, to remain in the Pallet Editor.
- 2 To return to the Pallet Editor at any time, press F1, BACK.
- 3 To set up position default information, press F2, POINT.
- 4 To record each position:
 - a Jog the robot to the position.
 - b Hold down the SHIFT key and press F4, RECORD.
- 5 When you are finished recording positions, press F5, DONE. You are returned to your program and the PALLETTIZING EX instruction has been inserted.

**6.5.5
PALLETIZING-END
Instruction**

The PALLETIZING-END instruction resets the pallet register after the robot has finished stacking or unstacking a pallet.

6.6 PALLET REGISTER INSTRUCTIONS

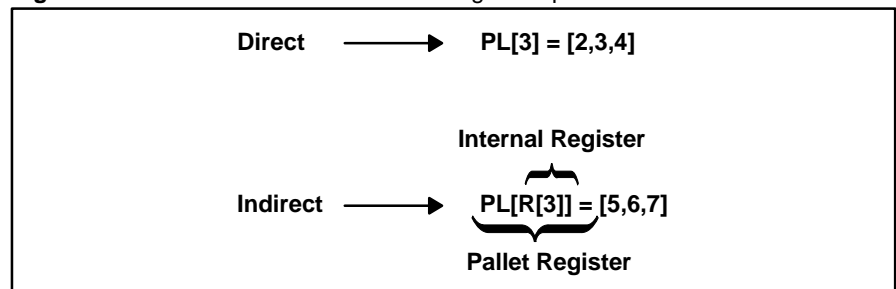
A pallet register stores layer, row, and column information for palletizing. Thirty-two registers are available for all the programs in the controller combined. Pallet registers are identified by numbers, 1-32. Pallet register instructions manipulate pallet registers arithmetically.

Pallet Register Addressing

Many instructions employ direct or indirect addressing techniques. When direct addressing is used, the actual value is entered into the instruction. For example, if the pallet register instruction $PL[2] = [2,3,4]$ is used, the contents of pallet register 2 is replaced with the value [2,3,4].

When indirect addressing is used, the instruction contains a register within a pallet register. This indicates that the actual value of the internal register becomes the pallet register number of the pallet register. See Figure 6-37.

Figure 6-37. Direct and Indirect Addressing Example



In Figure 6-37, the first instruction illustrates direct addressing. This instruction causes the contents of pallet register 3 to be replaced with the value [2, 3, 4].

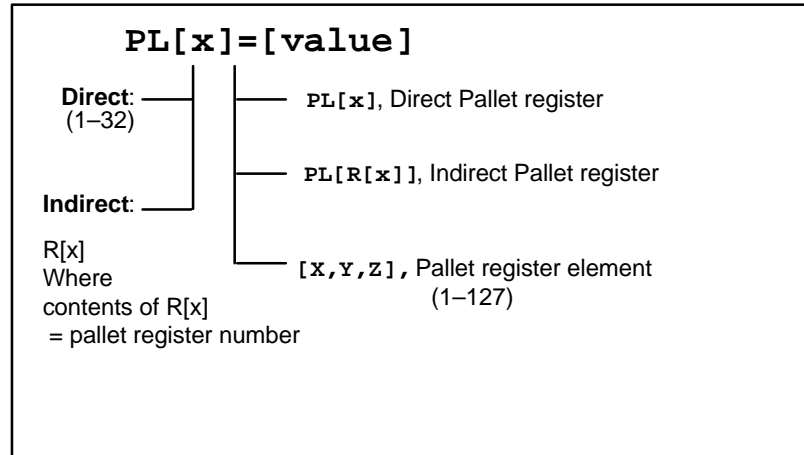
The second instruction in Figure 6-37 illustrates indirect addressing. In this instruction, $R[3]$ is the internal register and $PL[R[3]]$ is the external pallet register. If the value of the internal register $R[3]$ is 3, the pallet register number becomes $PL[R[3]=3]$ or $PL[3]$. Therefore, the result of the second instruction is that the contents of the pallet register, $PL[3]$, is to be replaced with the value [5, 6, 7].

PL[x] = [value]

The PL[x] = [value] instruction stores a value in a pallet register. See Figure 6-38.

PL[...] = ...

Figure 6-38. PL[x] = [value]



PL[x]=[value][operator][value]

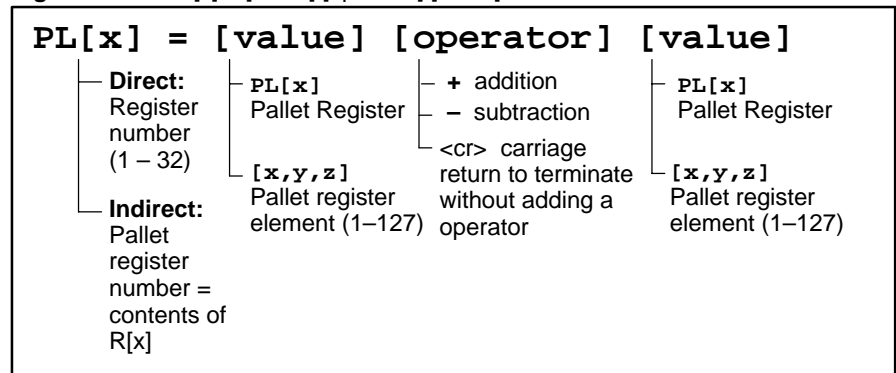
The PL[x] = [value] [operator] [value] instructions store the result of an arithmetic operation in a Pallet register. The arithmetic operations are addition and subtraction. See Figure 6-39.

You can use multiple arithmetic operators in a single instruction. However, there are the following limitations:

- You cannot mix +, -, or */ in the same instruction.
- The maximum number of arithmetic operators you can have in the same instruction is 5.

PL[...] = ... + ...
PL[...] = ... - ...

Figure 6-39. PL[x] = [value] [operator] [value]



For conditional branching instruction using PL[], refer to Section 6.10.3.

6.7 REGISTER INSTRUCTIONS

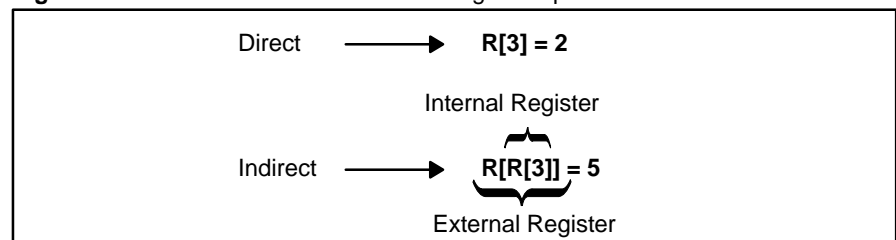
A register stores one number. A maximum of 256 registers are available for all the programs in the controller combined. The default number of registers is 32. Registers are identified by numbers. To increase the number of registers, perform a controlled start and select the PROGRAM INIT option from the controlled start menus. Register instructions manipulate register data arithmetically.

Register Addressing

Many instructions employ direct or indirect addressing techniques. When direct addressing is used, the actual value is entered into the instruction. For example, if the register instruction $R[2]=5$ is used, the current contents of register 2 is replaced with the value 5.

When indirect addressing is used, the instruction contains a register within a register. This indicates that the actual value of the internal register becomes the register number of the external register. See Figure 6-40.

Figure 6-40. Direct and Indirect Addressing Example



In Figure 6-40, the first instruction illustrates direct addressing. This instruction causes the current contents of register 3 to be replaced with the value 2.

The second instruction in Figure 6-40 illustrates indirect addressing. In this instruction, $R[3]$ is the internal register and $R[R[3]]$ is the external register. Since in the previous instruction the value of the internal register $R[3]$ is 2, the external register number becomes $R[R[3]=2]$ or $R[2]$. Therefore, the result of the second instruction is that the contents of the external register, $R[2]$, is to be replaced with the value 5.

R[x] = [value]

The R[x] = [value] instruction stores a value in a register. See Figure 6-41.

R[...] = ...

Figure 6-41. R[x] = [value]

R [x] = [value]	
Direct: (1-32)	AI[x] , Analog input signal Value of analog input signal x = contents of R[x]
	AO[x] , Analog output signal Value of analog output signal x = contents of R[x]
Indirect: R[x] Where contents of R[x] = register number	Constant value
	GI[x] , Group input signal Value of group input signal x = contents of R[x]
	GO[x] , Group output signal Value of group output signal x = contents of R[x]
	DI[x] , System digital input signal Value of system digital input signal x = contents of R[x]
	DO[x] , System digital output signal Value of system digital output signal x = contents of R[x]
	RI[x] , Robot digital input signal Value of robot digital input signal x = contents of R[x]
	RO[x] , Robot digital output signal Value of robot digital output signal x = contents of R[x]
	SI[x] , SOP input signal Value of SOP digital input signal x = contents of R[x]
	SO[x] , SOP output signal Value of SOP digital output signal x = contents of R[x]
	UI[x] , UOP input signal Value of UOP digital input signal x = contents of R[x]
	UO[x] , UOP output signal Value of UOP digital output signal x = contents of R[x]
	PR[x,y] , Position register element Contents of position register element x,y = contents of R[x]
	R[x] , Direct register
	R[R[x]] , Indirect register
	\$(system variable name)
	TIMER[x] , Timer value Value of program timer x = contents of R[x] The units of value are seconds.
	TIMER_OVERFLOW[x] , Timer overflow flag Contents of timer overflow flag x = contents of R[x] 0: Timer does not overflow. 1: Timer overflows
	NOTE: The result of the overflow is cleared when a timer reset instruction is executed.

R[x]=[value][operator][value]

The R[x] = [value] [operator] [value] instructions store the result of an arithmetic operation in a register. The arithmetic operations are

- Addition
- Subtraction
- Multiplication
- Division

- Whole number division (DIV)
- Remainder division (MOD)

See Figure 6-42.

You can use multiple arithmetic operators in a single instruction. However, there are the following limitations:

- You cannot mix +, -, or */ in the same instruction.
- The maximum number of arithmetic operators you can have in the same instruction is 5.

```
R[... ] = ... + ...
R[... ] = ... - ...
R[... ] = ... * ...
R[... ] = ... / ...
R[... ] = ... DIV ...
R[... ] = ... MOD ...
```

Figure 6-42. R[x] = [value] [operator] [value]

R[x] = [value] [operator] [value]			
Direct: Register number (1 - 32)	AI[x] Analog input	+ addition	AI[x] Analog input
	AO[x] Analog output	- subtraction	AO[x] Analog output
Indirect: Register number = contents of R[x]	Constant value	* multiplication	Constant value
	GI[x] Group input	/ division	GI[x] Group input
	GO[x] Group output	DIV whole number division	GO[x] Group output
	DI[x] System digital input	MOD remainder division	DI[x] System digital input
	RI[x] Robot digital input	<cr> carriage return to terminate without adding a operator	RI[x] Robot digital input
	DO[x] System digital output		DO[x] System digital output
	RO[x] Robot digital output		RO[x] Robot digital output
	R[x] Register		R[x] Register
	PR[x,y] Position register element		PR[x,y] Position register element
	SI[x] SOP Input		SI[x] SOP Input
SO[x] SOP Output		SO[x] SOP Output	
UI[x] UOP Input		UI[x] UOP Input	
UO[x] UOP Output		UO[x] UOP Output	
TIMER[x] Timer value		TIMER[x] Timer value	
TIMER_OVERFLOW[x] Timer overflow flag		TIMER_OVERFLOW[x] Timer overflow flag	

NOTE: The result of the overflow is cleared when a timer reset instruction is executed.

6.8 POSITION REGISTER INSTRUCTIONS

A position register stores positional information (x,y,z,w,p,r, configuration). A maximum of 64 position registers are available for all programs in the controller combined. The default number of position registers is 32. Position registers are identified by numbers, 1-32.

To increase the number of position registers, perform a controlled start and select the PROGRAM INIT option from the controlled start menus. Position register instructions manipulate position registers arithmetically. Refer to Appendix C for information on performing a controlled start.

There are two types of position register instructions:

- Instructions that manipulate position registers, PR[x]
- Instructions that manipulate position register elements, PR[i,j]

If your system is configured to have more than one group, you can set the group mask when you create any position register instruction. The group mask allows you to use function keys to specify:

- Whether the group mask will be used. If the group mask is not used, the position register instruction affects the default group only.
- The group or groups that the position register instruction will affect.

6.8.1 PR[x] Position Register Instructions

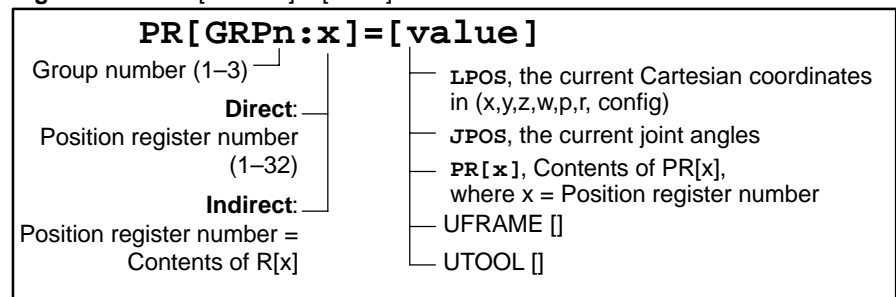
PR[GRPn:x] position register instructions manipulate the position register. They include assignment, addition, and subtraction instructions.

PR[GRPn:x] = [value]

The PR[GRPn:x] = [value] instruction stores positional information in a position register. See Figure 6-43.

PR[...] = ...

Figure 6-43. PR[GRPn:x] = [value]



PR[GRPn:x]=[value][operator][value]

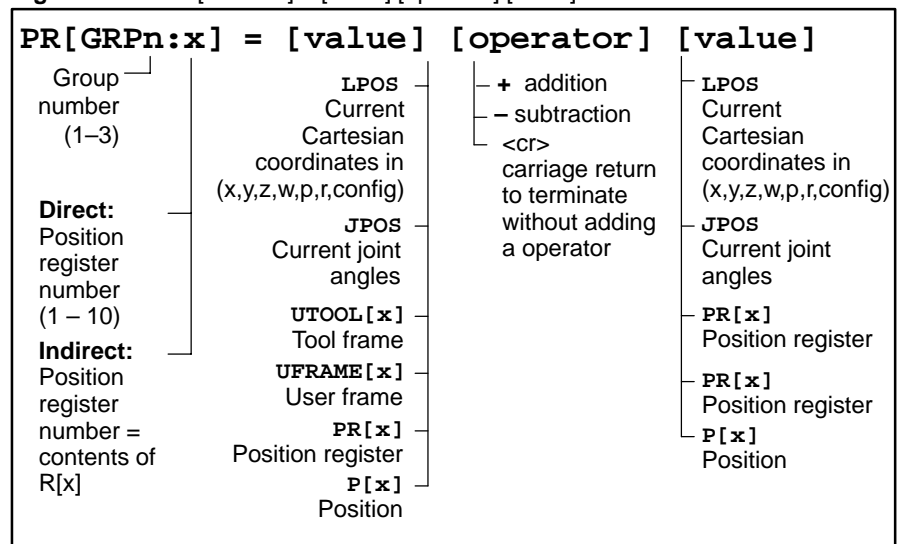
The PR[GRPn:x] = [value] [operator] [value] instructions store the result of an arithmetic operation in a register. The arithmetic operations are addition and subtraction. See Figure 6-44.

You can use multiple arithmetic operators in a single instruction. However, there are the following limitations:

- You cannot mix +, -, or */ in the same instruction.
- The maximum number of arithmetic operators you can have in the same instruction is 5.

```
PR[... ] = ... + ...
PR[... ] = ... - ...
```

Figure 6-44. PR[GRPn:x] = [value] [operator] [value]

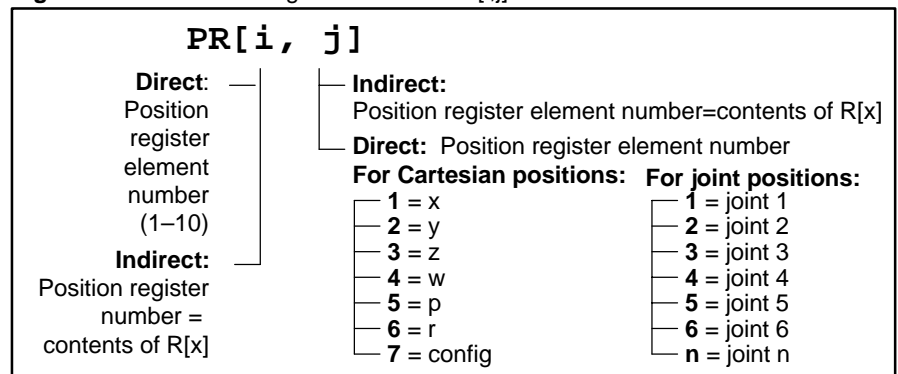


6.8.2 PR[i,j] Position Register Element Instructions

PR[i,j] position register element instructions manipulate a specific position register element. A position register element is one element of a specified position register. In the designation PR[i,j], the **i** represents the position register number and the **j** represents the position register element.

Position register element instructions include assignment, addition, and subtraction instructions. See Figure 6-45.

Figure 6-45. Position Register Element PR[i,j]

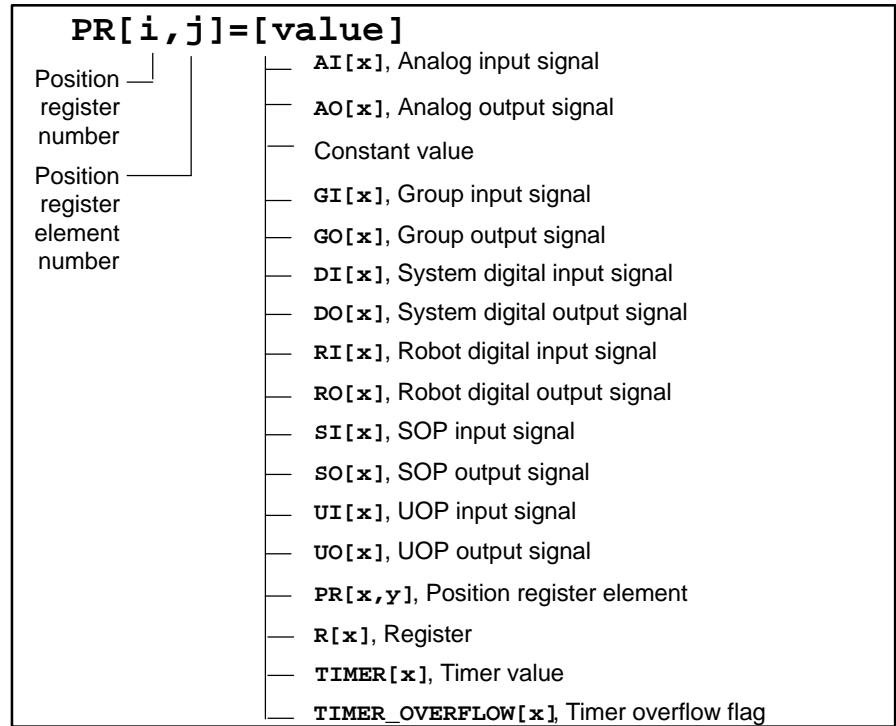


PR[i,j] = [value]

The PR[i,j] = [value] instruction stores positional information in a position register element. See Figure 6-46.

PR[...,...] = ...

Figure 6-46. PR[i,j] = [value]



PR[i,j]=[value][operator][value]

The PR[i,j] = [value] [operator] [value] instructions store the result of an arithmetic operation in a position register element. The arithmetic operations are addition, subtraction, multiplication, division, whole number division (DIV), and remainder division (MOD). See Figure 6-47.

You can use multiple arithmetic operators in a single instruction. However, there are the following limitations:

- You cannot mix +, -, or */ in the same instruction.
- The maximum number of arithmetic operators you can have in the same instruction is 5.

```
PR[ ...,... ]=...+...
PR[ ...,... ]=...-...
PR[ ...,... ]=...*...
PR[ ...,... ]=.../...
PR[ ...,... ]=...DIV...
PR[ ...,... ]=...MOD...
```

Figure 6-47. PR[i,j] = [value] [operator] [value]

PR[i,j]=[value] [operator] [value]			
Direct: Register number (1 - 32)	AI[x]	+ addition - subtraction * multiplication / division DIV whole number division MOD remainder division <cr> carriage return to terminate without adding a operator	AI[x]
	Analog input		Analog input
Indirect: Register number = contents of R[x]	AO[x]		AO[x]
	Analog output		Analog output
	Constant value		Constant value
	GI[x]		GI[x]
	Group input		Group input
	GO[x]		GO[x]
	Group output		Group output
	DI[x]		DI[x]
	System digital input		System digital input
	RI[x]		RI[x]
	Robot digital input		Robot digital input
	DO[x]		DO[x]
	System digital output		System digital output
	RO[x]		RO[x]
	Robot digital output		Robot digital output
	R[x]		R[x]
	Register		Register
	PR[x,y]		PR[x,y]
	Position register element		Position register element
	SI[x]		SI[x]
	SOP Input		SOP Input
	SO[x]		SO[x]
	SOP Output		SOP Output
	UI[x]		UI[x]
	UOP Input		UOP Input
	UO[x]		UO[x]
	UOP Output		UOP Output
	TIMER[x]		TIMER[x]
	Timer value		Timer value
	TIMER_OVERFLOW[x]		TIMER_OVERFLOW[x]
	Timer overflow flag		Timer overflow flag

6.9 INPUT/OUTPUT INSTRUCTIONS

Input/output, or I/O, instructions allow the program to turn on and off output signals and receive input signals. There are several kinds of I/O instructions:

- Digital input and output instructions
- Robot digital input and output instructions
- Analog input and output instructions
- Group input and output instructions
- PLC I/O (Optional)

6.9.1 Digital Input and Output Instructions

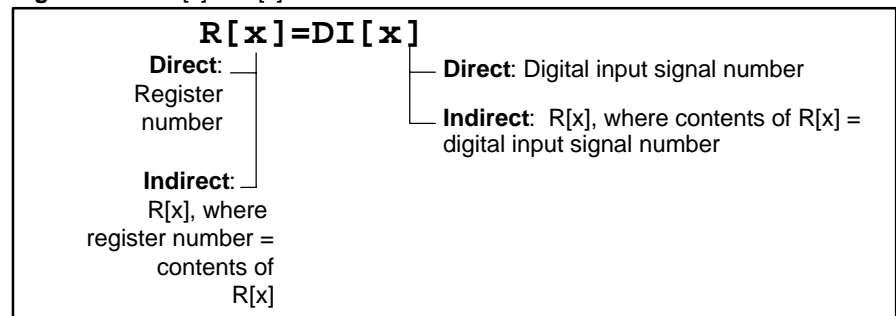
Digital input (DI) and digital output (DO) signals are user-controlled input and output signals. Use digital input and output instructions to control digital input and output signals in a program. Refer to Section 3.1.2.

R[x] = DI[x]

The R[x] = DI[x] instruction stores the condition of a digital input signal line (ON=1, OFF=0) in a register. See Figure 6-48.

R[] = DI[]

Figure 6-48. R[x] = DI[x]

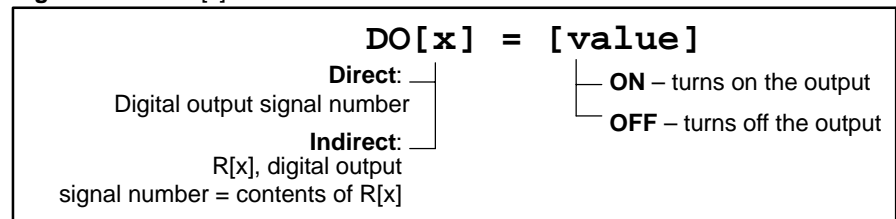


DO[x] = ON/OFF

The DO[x] = ON/OFF instruction turns on or off the specified digital output signal. See Figure 6-49.

DO[] = ...

Figure 6-49. DO[x] = ON/OFF

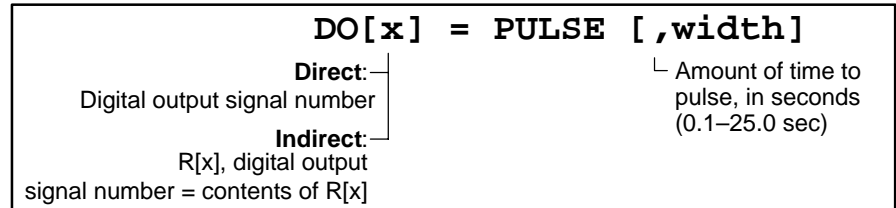


DO[x] = PULSE [,width]

The DO[x]=PULSE [,width] instruction turns on the digital output signal for the time specified. See Figure 6-50.

DO[] = ...

Figure 6-50. DO[x] = PULSE [,width]

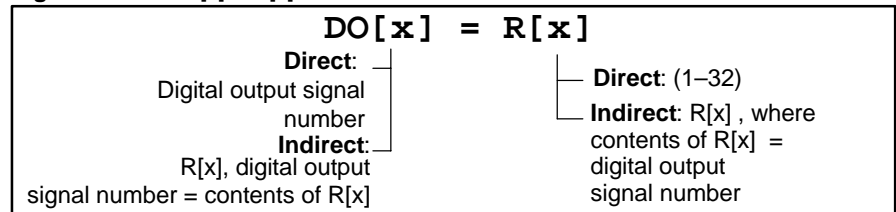


DO[x] = R[x]

The DO[x] = R[x] instruction turns on or off the specified digital output signal based on the value of the register. A value of 0 turns the specified digital output OFF. All values except zero turn the specified digital output ON. See Figure 6-51.

DO[] = ...

Figure 6-51. DO[x] = R[x]



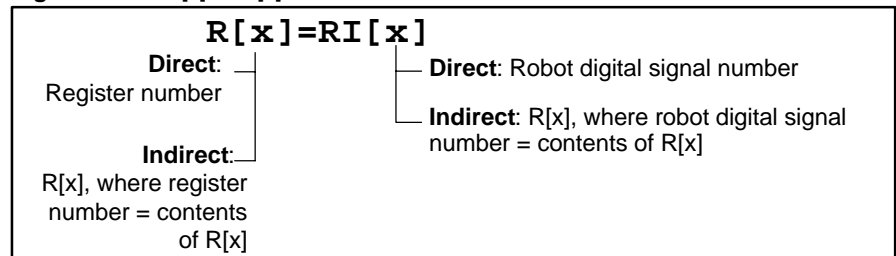
**6.9.2
Robot Digital Input and Output Instructions
R[x] = RI[x]**

Robot digital input (RI) and robot output (RO) signals are signals used to communicate between the controller and the robot. Refer to Section 3.1.4.

The R[x] = RI[x] instruction stores the condition of specified robot digital input signal (ON=1, OFF=0) in a register. See Figure 6-52.

R[] = RI[]

Figure 6-52. R[x] = RI[x]

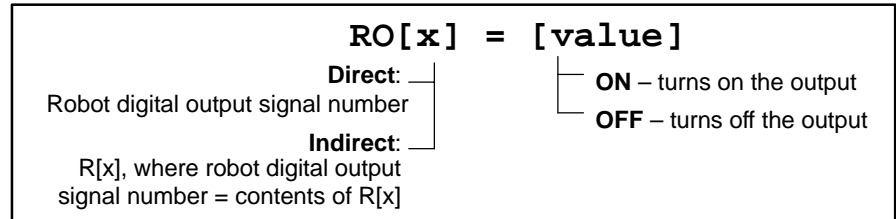


RO[x] = ON/OFF

The RO[x] = ON/OFF instruction turns on or off the specified robot digital output signal. See Figure 6-53.

RO[] = ...

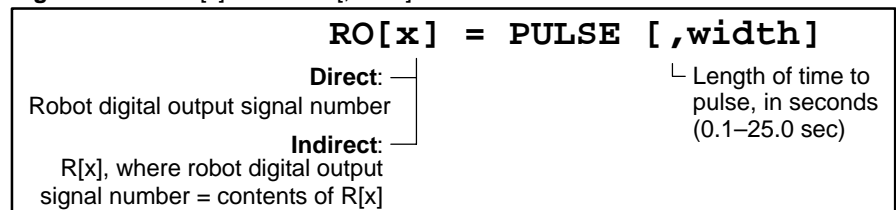
Figure 6-53. RO[x] = ON/OFF

**RO[x] = PULSE [,width]**

The RO[x]=PULSE [,width] instruction turns on the specified robot digital output signal for the time specified. See Figure 6-54.

RO[] = ...

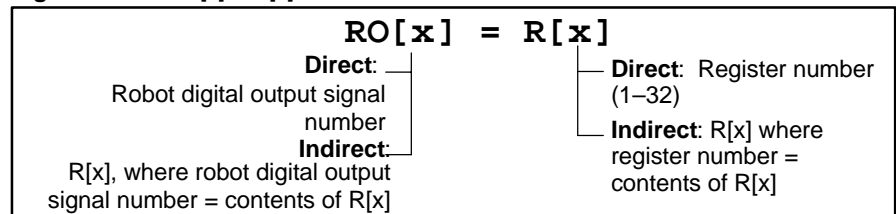
Figure 6-54. RO[x] = PULSE [,width]

**RO[x] = R[x]**

The RO[x] = R[x] instruction turns on or off the specified robot digital output signal based on the value of the register (1=ON, 0=OFF). See Figure 6-55.

RO[] = ...

Figure 6-55. RO[x] = R[x]



6.9.3 Analog Input and Output Instructions

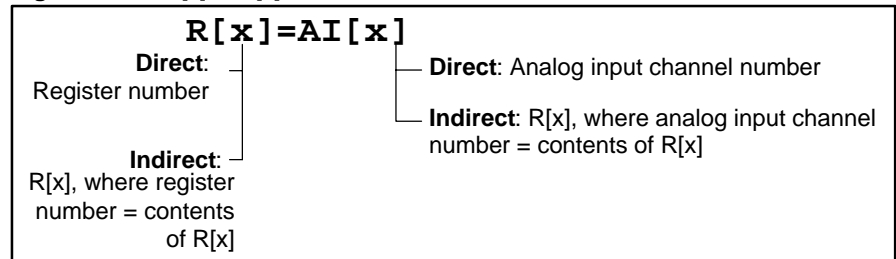
Analog input (AI) and analog output (AO) signals are continuous input and output signals whose magnitudes indicate data values, such as temperatures and voltages. Refer to Section 3.1.1.

R[x] = AI[x]

The R[x] = AI[x] instruction stores the value on an analog input channel in a register. See Figure 6-56.

R[] = AI[]

Figure 6-56. R[x] = AI[x]

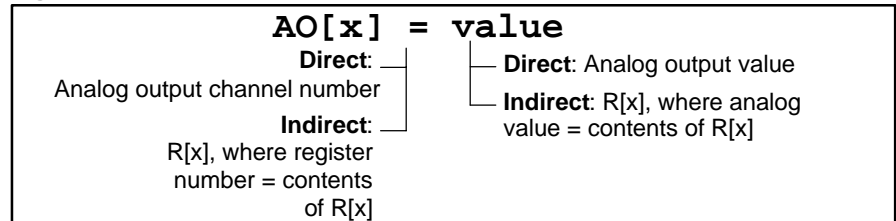


AO[x] = value

The AO[x]=value instruction sends a value on an analog output channel. See Figure 6-57.

AO[] = ...

Figure 6-57. AO[x] = value



6.9.4 Group Input and Output Instructions

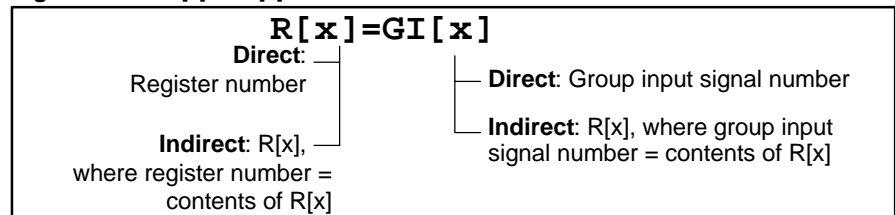
Group input (GI) and group output (GO) signals are several digital input and output signals that have been assigned to a group, can be read as a binary number, and can be controlled by one instruction. Refer to Section 3.1.3.

$R[x] = GI[x]$

The $R[x] = GI[x]$ instruction places the decimal value of the binary number on the specified group input into the specified register. See Figure 6-58.

$R[] = GI[]$

Figure 6-58. $R[x] = GI[x]$

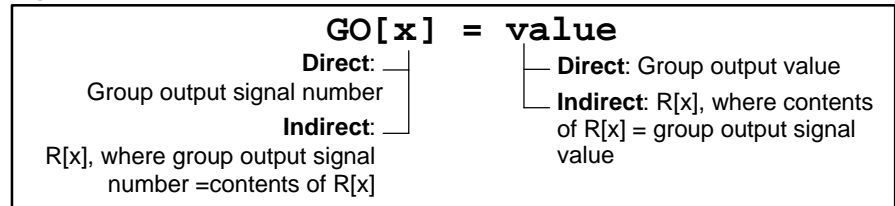


$GO[x] = \text{value}$

The $GO[x] = \text{value}$ instruction sends the binary equivalent of a value on the specified group output lines. See Figure 6-59.

$GO[] = \dots$

Figure 6-59. $GO[x] = \text{value}$



6.10 BRANCHING INSTRUCTIONS

Branching instructions cause the program to branch, or jump, from one place in a program to another. There are three kinds of branching instructions:

- Label definition instruction
- Unconditional branching instructions
- Conditional branching instructions
- Program end instruction

6.10.1 Label Definition Instruction

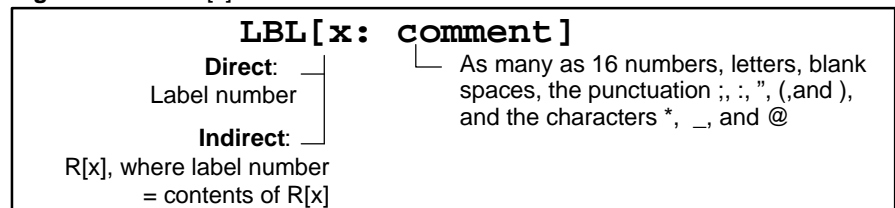
A label marks the location in a program that is the destination of a program branch. A label is defined using a label definition instruction.

A comment can be added to describe the label. After a label has been defined, it can be used with conditional and unconditional branching instructions. See Figure 6-60.

LBL[x]

LBL[. . .]

Figure 6-60. LBL[x]



6.10.2 Unconditional Branching Instructions

Unconditional branching instructions branch from one place in a program to another any time they are executed. There are two kinds of unconditional branching instructions:

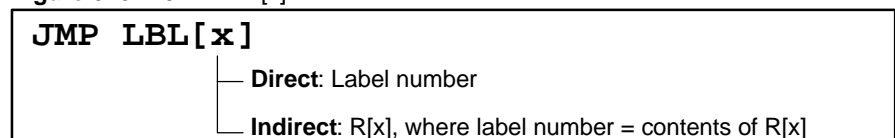
- **Jump instructions** – Cause the program to branch to a named label.
- **Subprogram call instructions** – Cause the program to branch to another program.

JMP LBL[x]

The JMP LBL[x] instruction causes the program to branch to the specified label. See Figure 6-61.

JMP LBL[. . .]

Figure 6-61. JMP LBL[x]



CALL program

The CALL program instruction causes the program to branch to another program and execute it. When the called program finishes executing, it returns to the main program at the first instruction after the call program instruction. See Figure 6-62.

CALL program

Figure 6-62. CALL Program Instruction



END

The program end instruction signals the end of a program. It is placed in the program automatically. See Figure 6-63.

Figure 6-63. Program End Instruction



6.10.3 Conditional Branching Instruction

Conditional branching instructions branch from one place to another in a program, depending on whether certain conditions are true. There are two kinds of conditional branching instructions:

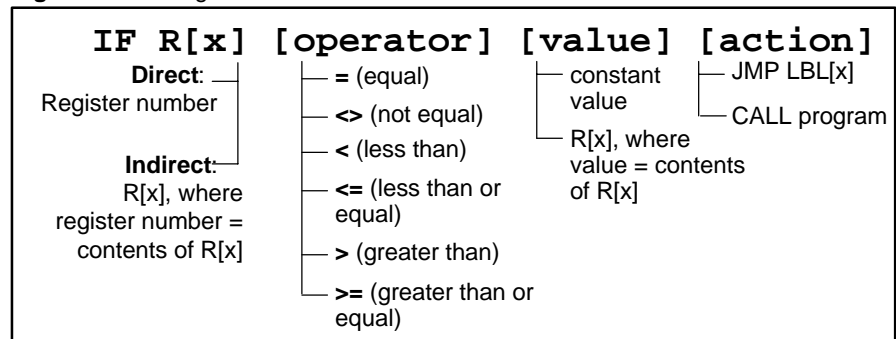
- **IF instructions** – Branch to a specified label or program if certain conditions are true. There are register IF instructions and input/output IF instructions.
- **SELECT instructions** – Branch to one of several jump or call instructions, depending on the value of a register.

IF R[x] [operator] [value] [action]

Register IF instructions compare the value contained in a register with another value and take an action if the comparison is true. See Figure 6-64.

IF R[...] =
 IF R[...] <>
 IF R[...] <
 IF R[...] <=
 IF R[...] >
 IF R[...] >=

Figure 6-64. Register IF Instruction



IF [I/O] [operator] [value] [action]

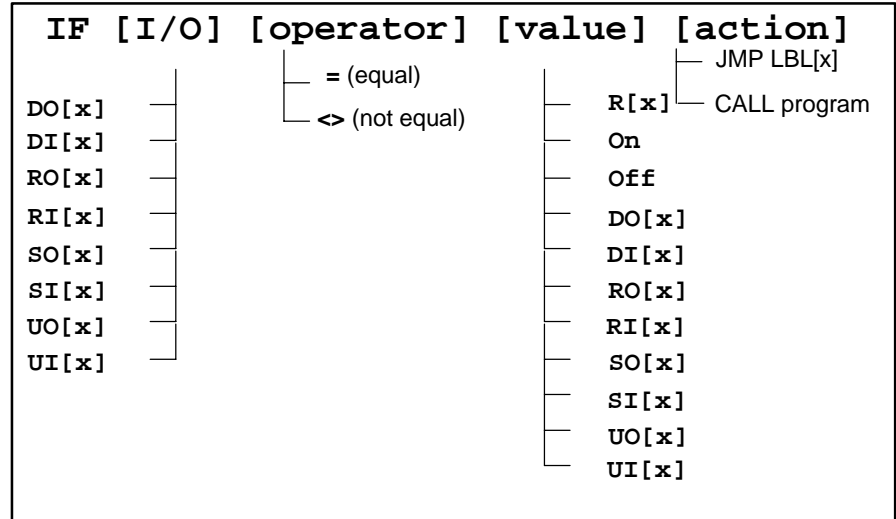
Input/output IF instructions compare an input or output value with another value and take an action if the comparison is true.

See Figure 6-65 to Figure 6-67.

Example

```
IF DI[...] = ... ..
IF DO[...] <> ... ..
```

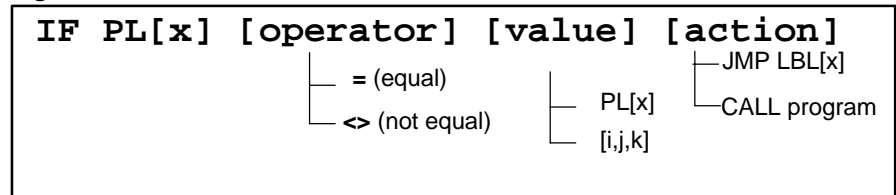
Figure 6-65. I/O IF Instruction for DI/DO, RI/RO, SI/SO and UI/UO



Example

```
IF PL[...] = ... ..
IF PL[...] <> ... ..
```

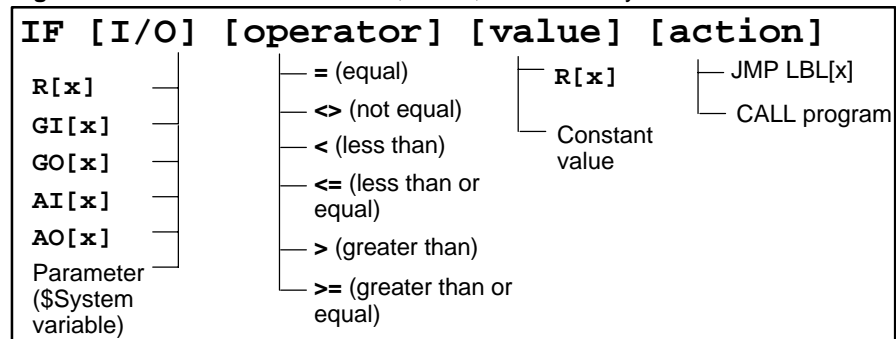
Figure 6-66. I/O IF Instruction for PL



Example

```
IF R[...] = ... ..
IF R[...] <> ... ..
IF R[...] < ... ..
IF R[...] <= ... ..
IF R[...] > ... ..
IF R[...] >= ... ..
```

Figure 6-67. I/O IF Instruction for R, AI/AO, GI/GO and System Variable



For an IF instruction, conditions can be connected using AND or OR, as follows:

- AND operator

IF [cond1] AND [cond2] AND ..., [action]

For example,

```
1: IF R[1]=1 AND R[2]=2 AND DI[2]=ON, JMP LBL[2]
```

- OR instruction

IF[cond1] OR [cond2] OR ..., [action]

For example,

```
1: IF DI[10]=ON OR R[7]=R[8], JMP LBL[2]
```

NOTE You cannot mix the AND and OR operators in the same operation.

When you replace the operator between AND and OR, any operators taught in the same line are also replaced automatically and the following message is displayed.

TPIF-062 AND operator was replaced to OR

TPIF-063 OR operator was replaced to AND

The maximum number of logical conditions that can be taught in the same operation is 5.

```
IF [cond1] OR [cond2] OR [cond4] OR [cond5], [action]
```

+-----+

Max 5 logical conditions

```
SELECT R[x] = [value1] [action]
             = [value2] [action]
             = [valueN] [action]
             ELSE [action]
```

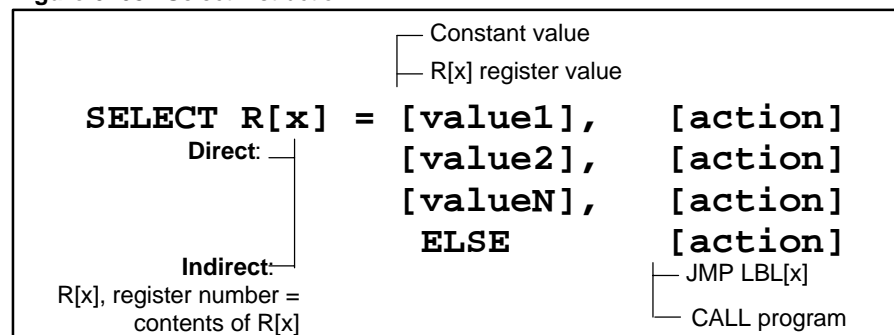
A select instruction compares the value of a register with one of several values and takes an action if the comparison is true:

- If the value of the register equals one of the values, the jump or call instruction associated with that value is executed.
- If the value of the register does not equal one of the values, the jump or call instruction associated with the word ELSE is executed.

See Figure 6-68.

```
SELECT R[ ] = ...
ELSE ...
```

Figure 6-68. Select Instruction



6.11 WAIT INSTRUCTIONS

Wait instructions delay program execution for a specified time or until a specified condition is true. When a wait instruction is executed, the robot does not execute any motion instructions. There are two kinds of wait instructions:

- **WAIT time** – delays program execution for a specified time.
- **WAIT condition** – delay program execution until specified conditions are true.

WAIT time

The WAIT time instruction delays program execution for a specified time. The time in a WAIT time instruction is specified in seconds, with a minimum unit of 0.01 seconds. See Figure 6-69.

WAIT ... (sec)

Figure 6-69. Wait Time

WAIT time

- **Direct:** Time in seconds, with a minimum unit of 0.01 seconds
- **Indirect:** R[x], where time in seconds = contents of R[x]

**WAIT [item] [operator] [value]
[time]**

WAIT condition instructions delay program execution until specified conditions are true or until an amount of time elapses (a timeout occurs). The timeout can be specified as one of the following:

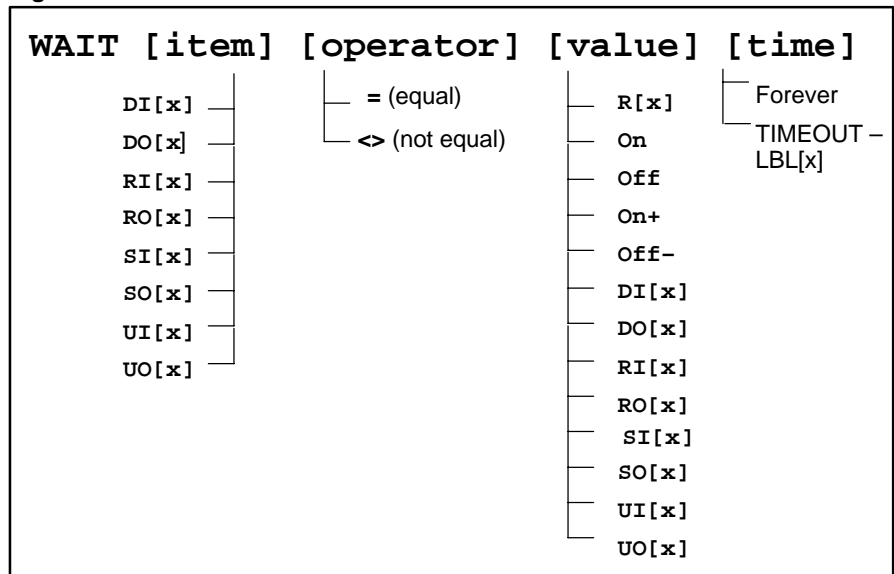
- **Forever** – the program will wait until the condition is true.
- **Timeout, LBL[i]** – the program will wait for the time specified in Timeout. If the condition is still not true, the program will branch to the specified label.

Specify the **timeout** by setting the system variable \$WAITTMOUT to a time, in milliseconds. The default timeout value is 3000 milliseconds. You can set \$WAITTMOUT using the parameter name instruction. Refer to Section 6.12.7 for information on the parameter name instruction.

See Figure 6-70 through Figure 6-72 for examples.

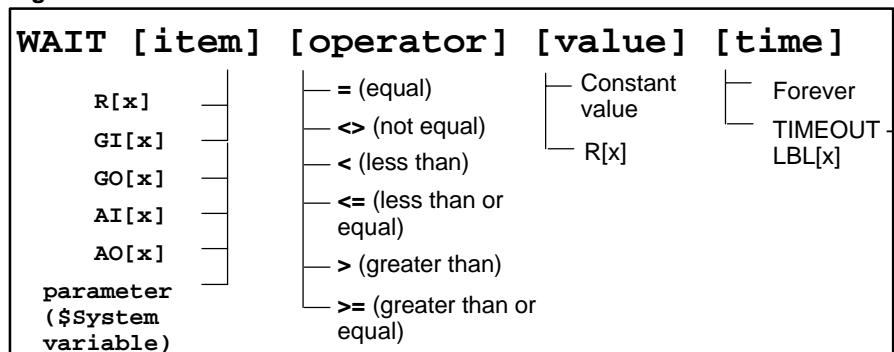
```
WAIT ... = ...
WAIT ... <> ...
```

Figure 6-70. WAIT Condition



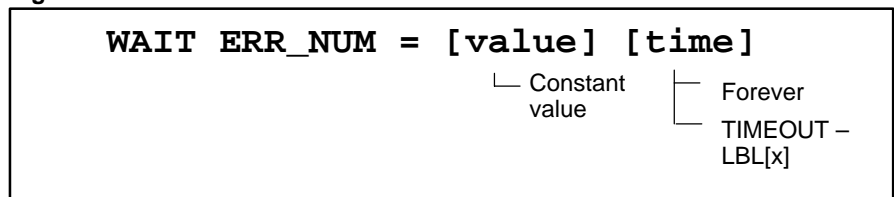
```
WAIT ... = ...
WAIT ... <> ...
WAIT ... < ...
WAIT ... <= ...
WAIT ... > ...
WAIT ... >= ...
```

Figure 6-71. WAIT Condition



```
WAIT ERR_NUM = ...
```

Figure 6-72. WAIT Condition



Error Number

Where:

ERR_NUM =aaabbb

aaa : Error facility code (decimal); Refer to Section A.1.1

bbb : Error number (decimal)

If 0 is specified as the error number “aaabbb” when an error occurs, the condition is satisfied.

For example, the instruction

WAIT ERR_NUM=11006, CALL PROG_A

Will cause the program PROG_A to be called when a “SRVO-006 HAND BROKEN” error occurs. (SRVO errors are facility code 11.)

Operators

For WAIT instructions, logical instruction editing can contain multiple logical statements connected by AND or OR operators.

- AND operator

WAIT [cond1] AND [cond2] AND ...

For example,

```
1: WAIT DI[1]=ON AND DI[2]=ON, TIMEOUT, LBL[1]
```

- OR instruction

WAIT [cond1] OR [cond2] OR ...

For example,

```
1: IF DI[10]=ON OR R[7]=R[8], JMP LBL[2]
```

NOTE You cannot mix the AND and OR operators in the same operation.

If an instruction contains multiple ORs or ANDs, and you change one of them, the others will also change. In this case, the following message is displayed:

```
TPIF-062      AND operator was replaced to OR
```

```
TPIF-063      OR operator was replaced to AND
```

The maximum number of logical condition; which can be taught in the same operation is 5. For example

```
WAIT [cond1] OR [cond2] OR [cond3] OR [cond4] OR [cond5]
```

```
+-----+
```

Max 5 logical conditions

6.12 MISCELLANEOUS INSTRUCTIONS

There are miscellaneous instructions for production control, user alarms, timer setting, speed override, program remarks, message handling, and parameter setting.

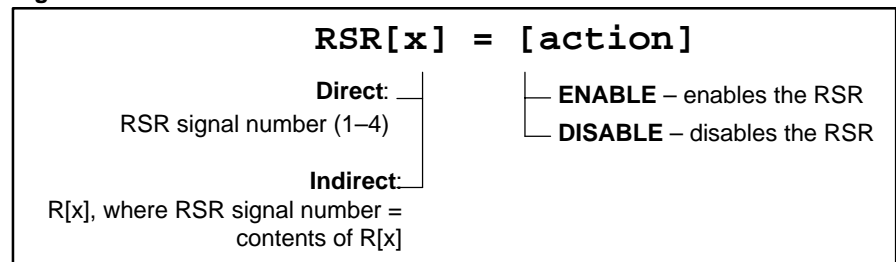
6.12.1 RSR Enable/Disable Instruction

RSR[x] = [action]

The RSR enable/disable instruction enables and disables the queueing process of the specified RSR. When an RSR signal is set to disable, the RSR signal will be ignored. See Figure 6-73.

RSR[. . .] = . . .

Figure 6-73. RSR Enable/Disable



6.12.2 User Alarm Instruction

UALM[x]

The user alarm instruction puts the program in an alarm condition, pauses the program, and causes a message to be displayed on the error message line as follows:

INTP 213 UALM[x] Message (prog_name, line_num)

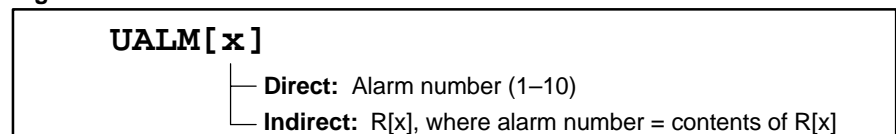
For example:

INTP 213 UALM[1] Check feeder (RSR001, 47)

If the program is resumed, program execution will continue from the next program line. The user alarm instruction specifies the alarm message to be displayed. Refer to Figure 6-74. and Section 3.17 for User Alarm Setup Screen.

UALM[. . .]

Figure 6-74. User Alarm



6.12.3 Timer Instruction

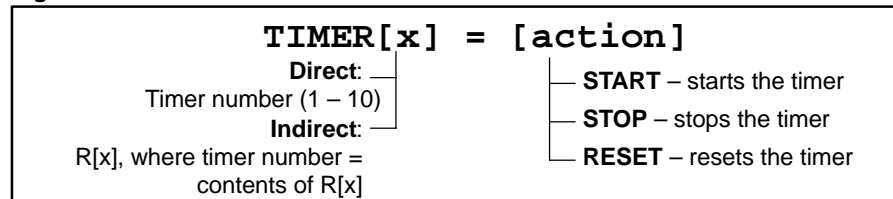
TIMER[x] = [action]

TIMER[...] = ...

Timer instructions allow you to start, stop, and reset up to ten different timers in a program. Timers allow you to determine how long a routine takes to execute, or how long your entire production program takes to execute. Timers can be started in one program and then stopped in another. The status of each timer is displayed on the Program Timer screen within the STATUS menu. See Figure 6-75.

You can display the status of program timers on the STATUS Prg Timer screen.

Figure 6-75. Timer

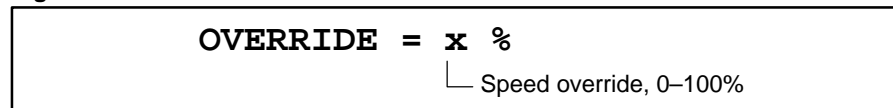


6.12.4 OVERRIDE Instruction

OVERRIDE = x %
OVERRIDE = 100%

The OVERRIDE instruction sets the speed override to a percentage value of the programmed speed. See Figure 6-76.

Figure 6-76. OVERRIDE



6.12.5 Remark Instruction

!remark text

The remark instruction allows you to annotate the program. Remark information does not affect the execution of the program. When you add a remark instruction, you enter the message to display within the program. The remark instruction can be from 1 to 32 alphabetic, numeric, punctuation, and blank space characters. The first character of a remark instruction is an exclamation point (!).

6.12.6 Message Instruction

MESSAGE [message content]

MESSAGE[...]

The MESSAGE instruction displays the specified message on the USER screen. The message can be from 1 to 23 alphabetic, numeric, punctuation, and blank space characters. If you want a blank line between messages, leave the message content empty. See Figure 6-77.

When the MESSAGE [message content] instruction is executed, the user screen is displayed automatically.

Figure 6-77. Message Instruction



6.12.7 Parameter Name Instruction

You can display and change the value of a system variable through the parameter name instruction, by using teach pendant read and write operations. Refer to Section 8.6 for more information on system variables.

NOTE Some system variables only allow you to display their value. Therefore, you might not be able to change the value of some system variables using the parameter name instruction.

Use Procedure 6-8 to define a parameter name instruction

There are two kinds of data types for a system variable:

- Numeric data type, which can be stored in a register.
- Position data type, which can be stored in a position register. There are three position data types possible:
 - Cartesian (XYZWPR)
 - Joint (J1 through J6)
 - Matrix (AONL)

When a position data type system variable is stored in a position register, the position register takes on the data type of the system variable.

NOTE If the system variable you are setting requires a BOOLEAN value (true or false), use 1 for TRUE and 0 for FALSE.

CAUTION

If you try to store a numeric system variable to a position register or a position system variable to a numeric register, you will receive an error message.

WARNING

System variables control how the robot and controller operate. Do not set system variables unless you are certain of their effect; otherwise, you could disrupt the normal operation of the robot and controller.

`$(parameter name) = [value]`

The `$(parameter name)=[value]` instruction allows you to change (write) the value of a system variable. See Figure 6-78.

`$...= ...`

Figure 6-78. Parameter Name Write Instruction

`$(parameter name) = [value]`

System variable name
limited to 30 characters

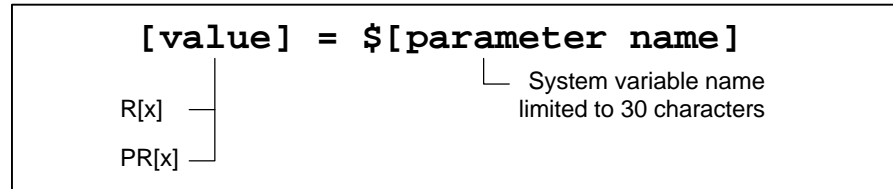
Value of system variable,
a number
R[x]
PR[x]

R[] / PR[] = \$[parameter name]

The [value] = \$[parameter name] instruction allows you to display (read) the value of a system variable. See Figure 6-79.

... = \$...

Figure 6-79. Parameter Name Read Instruction



Procedure 6-8 Defining a Parameter Name Instruction

- Condition** ■ You are currently editing a teach pendant program.
- Step** 1 Move the cursor to the line number where you want to a parameter name instruction.
- 2 Press F2, [INST]. You will see a screen similar to the following.

```

Instruction JOINT 10%
1 Miscellaneous      5 MACRO
2 Skip              6 Tool_Offset
3 Offset            7 LOCK_PREG
4 Program control   8 --next page--
ABC
1/2
  1:
[END]
Select item
[SELECT]

```

- 3 Select Miscellaneous. You will see a screen similar to the following.

```

Miscellaneous stat JOINT 10%
1 RSR [ ]          5 Remark
2 UALM [ ]         6 MESSAGE
3 TIMER [ ]        7 Parameter name
4 OVERRIDE         8 --next page--
ABC
1/2
  1:
[END]

```

4 Select Parameter name. You will see a screen similar to the following.

- 1 = System variable write operation \$...=...
- 2 = System variable read operation ...=\$...

IF statement		JOINT 10%
1	\$...=...	5
2	...=\$...	6
3		7
4		8

5 If you select 1 to change (write) to a system variable using the parameter name instruction

- a You will see a screen similar to the following.

ABC	LINE 0
ABC	1/2
1: \$ [] = [] .	
[END]	

- b Press ENTER to begin entering the system variable name. Enter the system variable name. Press ENTER when you are finished entering the name. You will see a screen similar to the following.

Miscellaneous stat		JOINT 10%
1	R []	5
2	Constant	6
3	PR[]	7
4		8
ABC		1/2
1: \$...= []		
[END]		

- c Select the data type from which you want to get the data to store in the system variable.
- d Enter the value of the constant or register number and press ENTER.

6 If you select 2 to display (read) a system variable using the parameter name instruction

a You will see a screen similar to the following.

```

Miscellaneous stat          JOINT 10%
1 R [ ]                    5
2 PR[ ]                    6
3                           7
4                           8
ABC
                               1/2
1 [ ] =$...
[END]

```

b Select the data type to which you want to store the value of a system variable.

c Enter the value of the constant or register number and press ENTER. You will see a screen similar to the following.

```

1: R[1] =$ [ ]

```

d Press ENTER to begin entering the system variable name. Enter the system variable name. Press ENTER when you are finished entering the name.

7 The following screen shows both a read and write parameter name instruction.

```

ABC          JOINT 10%
                               1/2
1: $( System variable name )
: =Constant
2: R[1]=
: $( System variable name )
[END]

```

6.12.8 Maximum Speed Instruction

The maximum speed instructions set the maximum speed of joint motion and linear or circular motion in the program. If the motion speed exceeds the value designated by this instruction, the motion speed is limited by the designated value.

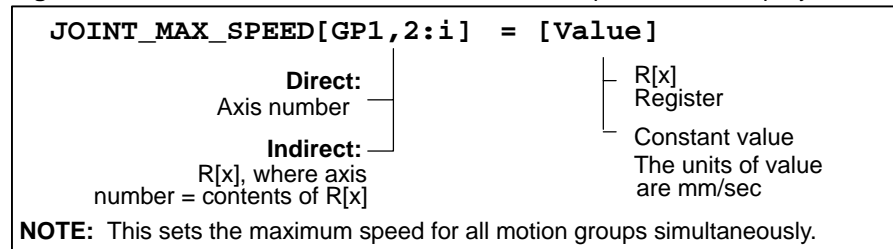
If you use a maximum speed instruction and

- If a macro program is called, the maximum speed value is set back to the default value.
- If a called macro program sets the maximum speed, the maximum speed value is set back to the default value when returning to the calling program.

Figure 6-80 and Figure 6-81 show the maximum speed instructions used in a multiple motion group system.

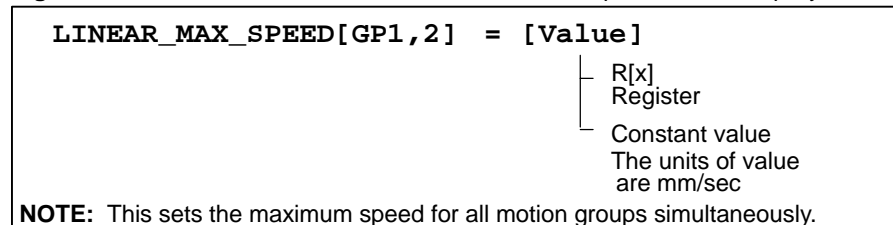
JOINT_MAX_SPEED[...] = ...

Figure 6-80. JOINT_MAX_SPEED Instruction – Multiple Motion Group Syntax



LINEAR_MAX_SPEED[...] = ...

Figure 6-81. LINEAR_MAX_SPEED Instruction – Multiple Motion Group Syntax



The maximum speed instructions have been shown as they would be used in a multiple motion group system. The syntax for the commands is shown in Figure 6-82 and Figure 6-83 when they are used in a single motion group system.

Figure 6-82. JOINT_MAX_SPEED Instruction – Single Motion Group Syntax

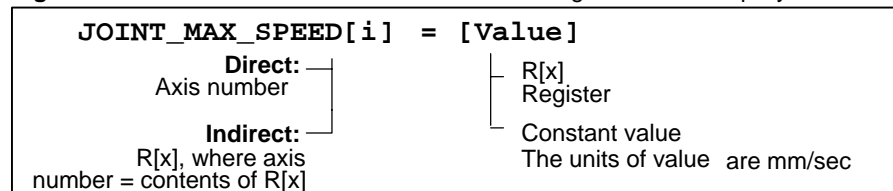
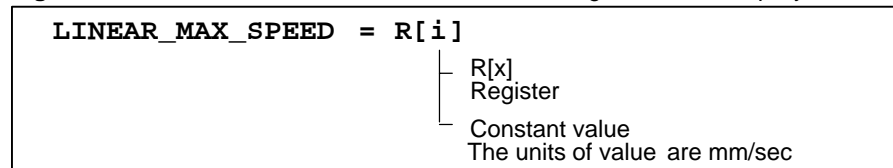


Figure 6-83. LINEAR_MAX_SPEED Instruction – Single Motion Group Syntax



6.13 SKIP INSTRUCTION

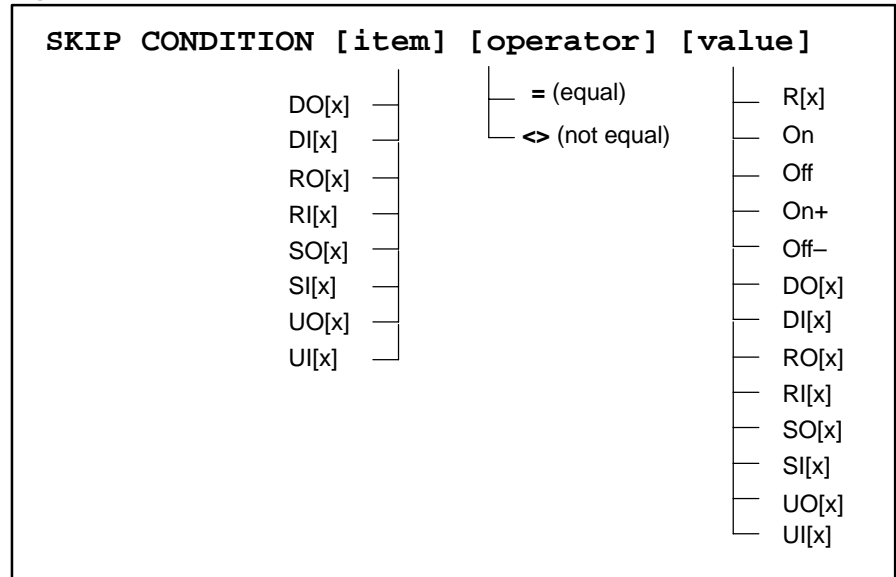
SKIP CONDITION [/O] = [VALUE]

SKIP CONDITION ...=...
SKIP CONDITION ...<>...

The skip instruction sets the conditions for executing robot motion when using the skip motion option in a motion instruction. These conditions are true until they are reset by another skip instruction. Refer to Section 6.3.6 for more information.

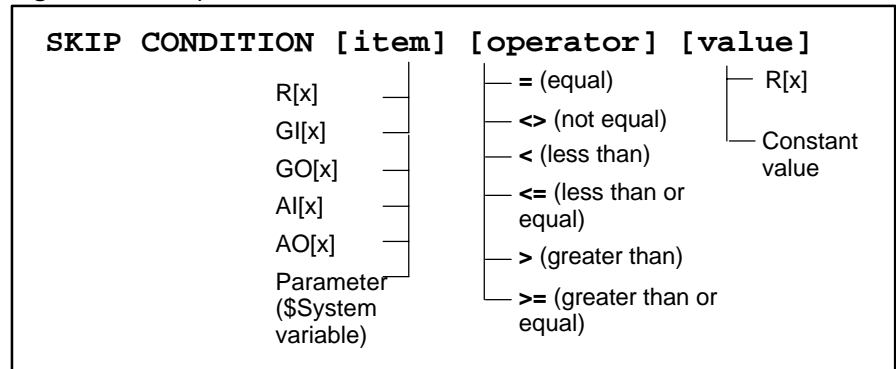
See Figure 6-84 to Figure 6-86.

Figure 6-84. Skip Condition



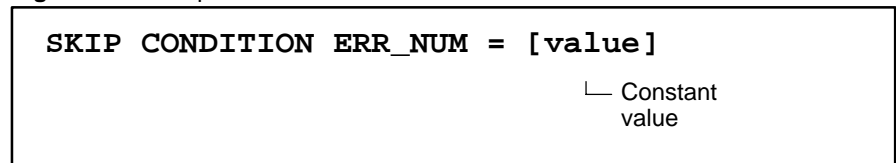
SKIP CONDITION ...=...
SKIP CONDITION ...<>...
SKIP CONDITION ...<...
SKIP CONDITION ...<=...
SKIP CONDITION ...>...
SKIP CONDITION ...>=...

Figure 6-85. Skip Condition



SKIP CONDITION ERR_NUM=...

Figure 6-86. Skip Condition



Error Number

ERR_NUM =aaabbb

aaa : Error ID (decimal); Refer to Section A.1.1

bbb : Error number (decimal)

If 0 is specified as error number “aaabbb,” when any kind of error occurs, the condition is satisfied.

For example,

```
SKIP CONDITION ERR_NUM=11006
```

This specifies the “SRVO-006 Hand broken” error because SRVO ID number is 11.

Operators

For the SKIP instruction, you can connect conditions using AND or OR operators, as follows:

- AND operator

```
SKIP CONDITION [cond1] AND [cond2] AND ...
```

For example,

```
1: SKIP CONDITION R[1]=1 AND R[2]=2
```

- OR instruction

```
SKIP CONDITION [cond1] OR [cond2] OR ...
```

For example,

```
1: IF DI[10]=ON OR R[7]=R[8], JMP LBL[2]
```

```
1: SKIP CONDITION R[1]=1 OR R[2]=2
```

NOTE You cannot mix AND and OR in the same operation.

If you replace the operator between AND and OR, any other operators taught in the same line are also replaced automatically and the following message is displayed:

```
TPIF-062      AND operator was replaced to OR
```

```
TPIF-063      OR operator was replaced to AND
```

The maximum number of logical conditions that can be taught in the same operation is 5.

```
SKIP CONDITION [cond1] OR [cond2] OR [cond3] OR [cond4] OR [cond5]
```

```
+-----+
```

Max 5 logical conditions

6.14 OFFSET INSTRUCTIONS

Offset instructions specify positional offset information or the frames used for positional information. There are five offset instructions:

- **Positional offset condition** – contains information on the distance or degrees to offset positional information

CAUTION

Recorded positions are not affected by UFRAME and UFRAME has no affect during playback. However, **position registers are recorded with respect to UFRAME**. If you change UFRAME, any recorded position registers will also change.

- **User frame**
 - Sets the number of the user frame to use
 - Defines a user frame
- **Tool frame**
 - Sets the number of the tool frame to use
 - Defines a tool frame

If your system is configured to have more than one group, you can set the group mask when you create any offset instruction that contains a position register. The group mask allows you to use function keys to specify:

- Whether the group mask will be used. If the group mask is not used, the position register will affect the default group only.
- The group or groups that the position register will affect.

OFFSET CONDITION PR[x] item

The OFFSET CONDITION PR[x] item instruction specifies a position register that contains the offset information used when the OFFSET command is executed. When a user frame is specified in UFRAME[y], that user frame is used when the offset command uses the offset specified in PR[x]. The OFFSET command is entered in the motion instruction. Refer to Section 6.3.6 for more information. See Figure 6-87.

OFFSET CONDITION ...

Figure 6-87. Offset Condition

OFFSET CONDITION PR[x], item	
Direct: Position register number	} no item UFRAME[y]
Indirect: R[x], where position register number = contents of R[x]	

UFRAME_NUM = [value]

The UFRAME_NUM=[value] instruction sets the number of the user frame to use. A value of zero indicates that no user frame is used. This means that world frame is used. See Figure 6-88. Refer to Section 3.8.2 for setting up the user frame.

NOTE To verify that this feature is enabled, check the value of \$USEUFRAME and be sure it is set to TRUE.

NOTE This instruction can be used only if your system has the *User frame input* option installed.

UFRAME_NUM = ...

Figure 6-88. UFRAME_NUM=[value]

UFRAME_NUM[GRP1,2,3] = [value]	
┌	Direct: User frame number (0-5)
└	Indirect: R[x], where user frame number = contents of R[x]

UTOOL_NUM = [value]

The UTOOL_NUM=[value] instruction sets the number of the tool frame to use. A value of zero indicates that no tool frame is used. This means that the frame defined by the faceplate coordinates is used. See Figure 6-89. Refer to Section 3.8.1 for setting up the tool frame.

UTOOL_NUM = ...

Figure 6-89. UTOOL_NUM=[value]

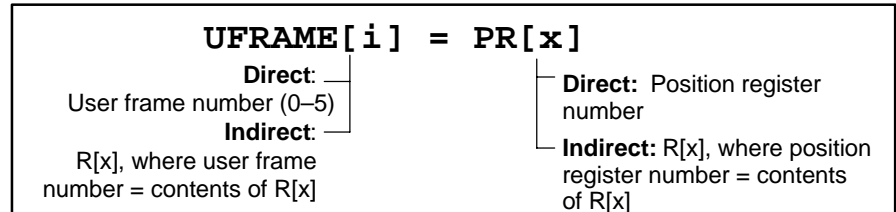
UTOOL_NUM[GRP1,2,3] = [value]	
┌	Direct: tool frame number (0-5)
└	Indirect: R[x], where tool frame number = contents of R[x]

UFRAME[i] = PR[x]

The UFRAME[i] = PR[x] instruction defines the specified user frame using the information contained in a position register. See Figure 6-90.

UFRAME[] = ...

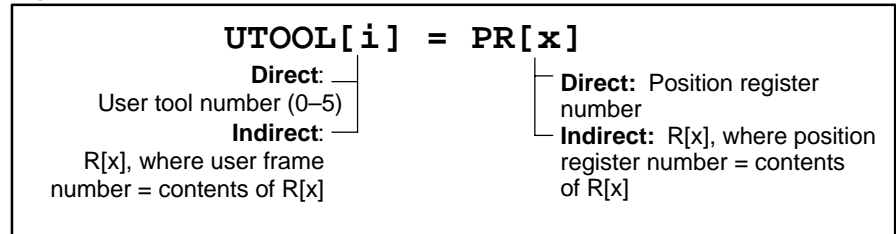
Figure 6-90. UFRAME[i] = PR[x]

**UTOOL[i] = PR[x]**

The UTOOL[i] = PR[x] instruction defines the specified tool frame using the information contained in a position register. See Figure 6-91.

UTOOL[] = ...

Figure 6-91. UTOOL[i] = PR[x]



6.15 TOOL OFFSET CONDITION INSTRUCTION

A tool offset condition instruction specifies the offset condition used in a tool offset instruction. Execute a tool offset condition instruction before executing the corresponding tool offset instruction. After the tool offset conditions have been specified, they remain in effect until the program terminates or the next tool offset condition instruction is executed. See the following screen for an example.

Instruction	JOINT 30%
1 Miscellaneous	Tool_Offset
2 Program control	6 MACRO
3 Skip	7
4 Offset	8 ---next page---

program

- The position register specifies the direction in which the target position shifts, as well as the amount of shift.
- The tool coordinate system is used for specifying offset conditions.
- When the number of a tool coordinate system is not specified, the currently selected tool coordinate system is used.
- When the position data is given as joint coordinates, an alarm is issued and the program pauses.

A tool offset instruction moves the robot from the target position (that was originally recorded in the position data) to another position by the amount specified using the tool offset condition instruction. The tool offset condition instruction specifies the condition when the offset is applied. See Figure 6-92.

Figure 6-92. Tool Offset Condition Instruction

TOOL_OFFSET CONDITION PR[i] (UTOOL[j])	
Position register number (1 to 10)	┌ └
	Tool frame number (1 to 5)

Example

```
1: TOOL_OFFSET PR[1]
2: J P[1] 100% FINE
3: L P[2] 500mm/sec FINE Tool_Offset
```

6.16 MULTIPLE CONTROL INSTRUCTIONS

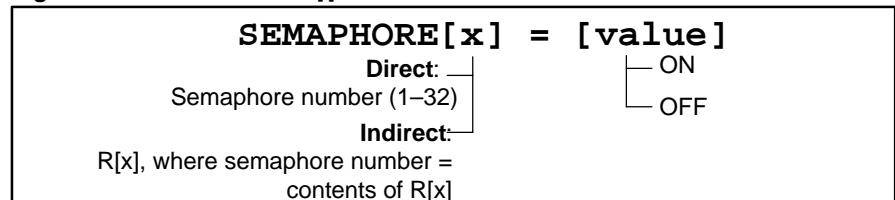
Multiple control instructions are used for multi-tasking. Multi-tasking allows you to execute more than one task at a time.

SEMAPHORE[x] = ON/OFF

The SEMAPHORE[x] = ON/OFF instruction sets the semaphore number to on or off. Semaphores are used in multi-tasking to start or delay a second program. See Figure 6-93.

SEMAPHORE[...] = ...

Figure 6-93. SEMAPHORE[i] = ON/OFF

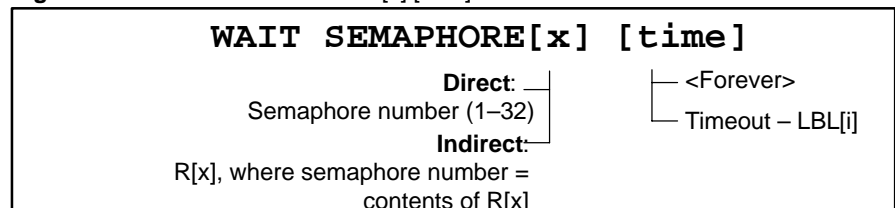


WAIT SEMAPHORE[x] [time]

The WAIT SEMAPHORE[x] [time] instruction suspends its program execution until any program that is currently executing reaches the line that contains the specified semaphore and that semaphore is set to ON. This instruction can delay its program execution forever, or for a specified time. See Figure 6-94.

WAIT SEMAPHORE[...]

Figure 6-94. WAIT SEMAPHORE[x] [time]



RUN program

The RUN program instruction causes the selected program to immediately begin to execute. See Figure 6-95.

RUN program

Figure 6-95. RUN program



6.17 MACRO COMMAND INSTRUCTION

The macro command instruction specifies the macro command to be executed when the program is run. A macro command is a separate program that contains a series of instructions to perform a task.

You can define as many as 20 macro commands, depending on how your system was set up. When a macro command is defined, it is available to all programs. See Figure 6-96.

`Macro_program_name`

Figure 6-96. Macro Command Instruction



`Macro_program_name`

Refer to Chapter 3 for information on setting up macro commands.

The HandlingTool software provides six predefined macro commands. These are:

- Open hand 1
- Close hand 1
- Relax hand 1
- Open hand 2
- Close hand 2
- Relax hand 2

Refer to Section 3.10, “Macro Commands,” for information on setting up and executing macro commands.

6.18 PROGRAM CONTROL INSTRUCTIONS

Program control instructions direct program execution. Use these when you want areas of your program to pause, abort, resume a program, and handle errors.

6.18.1 PAUSE Instruction

A PAUSE instruction suspends program execution in the following manner:

- Any motion already begun continues until completed.
- All connected timers continue being incremented.
- All PULSE instructions that are currently running continue to run until they are completed.
- Any instruction that is currently running, except program call instructions, is completed. Program call instructions are performed when the program is resumed.

See Figure 6-97.

PAUSE

Figure 6-97. PAUSE

PAUSE

6.18.2 ABORT Instruction

An ABORT instruction ends the program and cancels any motion in progress or pending. After an ABORT instruction is executed, the program cannot continue; it must be restarted. See Figure 6-98.

ABORT

Figure 6-98. ABORT

ABORT

6.18.3 Error Program Instruction

The error program instruction defines the program name that will be stored in the system variable \$ERROR_PROG. The use of the \$ERROR_PROG system variable varies depending on how your system is set up. See Figure 6-99.

ERROR_PROG = program

ERROR_PROG = ...

Figure 6-99. Error Program

ERROR_PROG = program

└ Name of program to be run
(1-8 characters)

6.18.4 Resume Program Instruction

The resume program instruction defines the program name that will be stored in the system variable \$RESUME_PROG. The use of the \$RESUME_PROG system variable varies depending on how your system is set up. See Figure 6-100.

RESUME_PROG = program

RESUME_PROG = ...

Figure 6-100. RESUME_PROG = program

RESUME_PROG = program

└ Name of program to run
(1-8 characters)

6.19 SENSOR INSTRUCTIONS

Sensor instructions send and receive information from external sensors through the RS-232-C serial port. The R-J2 controller can receive position offset data and transformation data from any sensory device that is capable of handling the R-J2 sensor protocol.

You must have the FANUC sensor interface option to use sensor instruction. Refer to Chapter 11 for more information about how to set up and use the FANUC Robotics sensor interface. There are two kinds of sensor instructions:

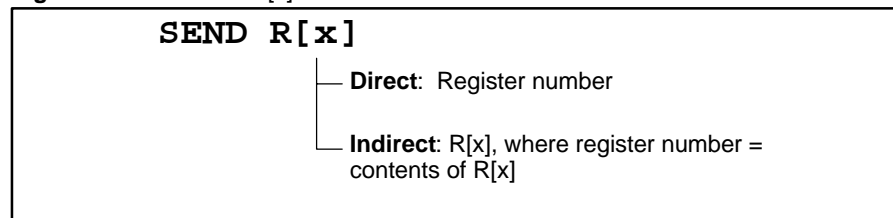
- Send sensor instructions
- Receive sensor instructions

SEND R[x]

The SEND sensor instruction notifies the sensor port which register will be used to terminate communication. See Figure 6-101.

SEND R[]

Figure 6-101. SEND R[x]



RCV R[x] LBL[x]

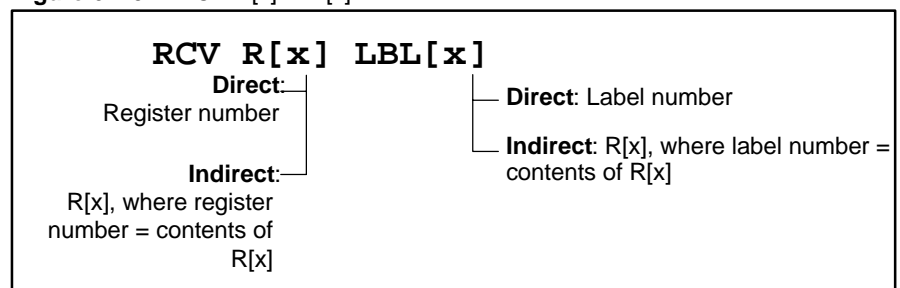
The receive sensor instruction waits until the register specified by the SEND instruction indicates that all information has been received. It then terminates the connection and continues the program execution.

If an error occurs (negative register value), then the program branches to LBL[x]. If the value of R[x] does not change, it waits forever.

See Figure 6-102.

RCV R[] LBL[]

Figure 6-102. RCV R[x] LBL[x]



RCV R[x] LBL[x] TIMEOUT, LBL[y]

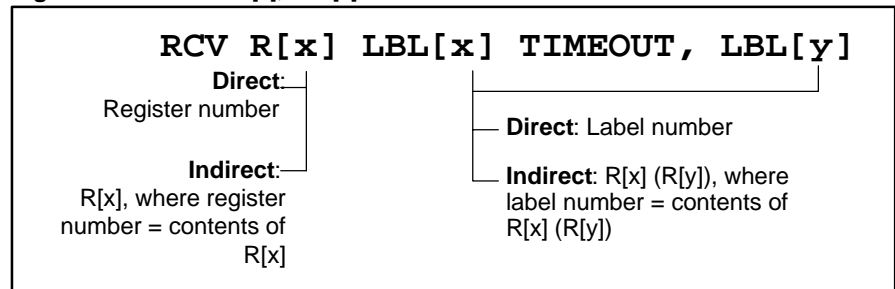
The receive sensor branch instruction waits until the register specified by the SEND instruction indicates that all information has been received. It then terminates the connection and continues the program execution.

If an error occurs (negative register value), then the program branches to LBL[x]. If the value of R[x] does not change, it waits until the value specified in TIMEOUT and then branches to LBL[y].

See Figure 6-103.

RCV R[] LBL[] TIMEOUT, LBL[]

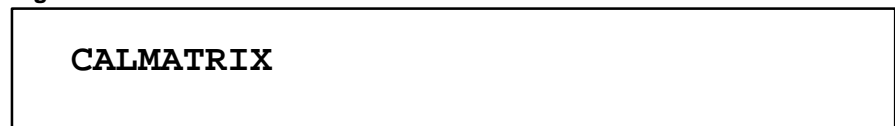
Figure 6-103. RCV R[x], LBL[x]

**CALMATRIX**

The CALMATRIX instruction computes the transformation matrix for offset data. The transformation matrix is computed based on actual taught data points and the measured data of the three point data stored in the position registers. The transformation matrix is stored in the positional data specified by the \$SENS_IF[] system variables. See Figure 6-104.

CALMATRIX

Figure 6-104. CALMATRIX Instruction



6.20 MOTION GROUP INSTRUCTIONS

Motion group instructions allow you to program multiple motion groups independently of each other.

Motion group instruction can be used to specify the:

- Motion type for individual groups (except for circular motions)
- Travel speed for individual groups
- Termination type for individual groups

There are two types of motion group instructions:

- Independent motion group instruction Independent GP, Section 6.20.1
- Simultaneous motion group instruction Simultaneous GP, Section 6.20.2

To define an Independent or Simultaneous motion group instruction in a teach pendant program, refer to Section 6.20.3.

6.20.1 Independent Motion Group Instructions

With an Independent motion group instruction, each motion group operates with a separately taught motion type, speed, and termination type.

The following example shows independent motion for motion groups 1 and 3, with motion mask [1,*,1,*,*].

```

Independent GP
GP1 L P[1] 90mm/s CNT 100
GP3 J P[1] 100% CNT 50

```

6.20.2 Simultaneous Motion Group Instructions

With a Simultaneous motion group instruction, each motion group operates with a separately taught motion type, but with the same time (the longest travel time among all groups based on the programmed speeds). The termination type for the motion group with the smallest CNT value (near the FINE value) is also used for the other motion groups.

The following example program shows simultaneous motion for motion groups 1 and 3, with motion mask [1,*,1,*,*]. The travel time for the motion group with the longest travel time is also used for the other motion groups. The programmed travel speeds might not always be used. The termination type value for group 3 (GP3), with the smallest CNT value, is also applied to group 1 (GP1).

```

Simultaneous GP
GP1 L P[1] 90mm/s CNT 100
GP3 J P[1] 100% CNT 50

```

6.20.3 Defining Motion Group Instructions

Procedure 6-9 describes how to define an Independent or Simultaneous motion group instruction. The example program specifies motion groups 1 and 3, with motion mask [1,*,1,*,*].

NOTE You cannot define an Independent or Simultaneous motion group instruction for a circular motion statement or a control statement.

Procedure 6-9 Defining Motion Group Instructions

- Condition** ■ You cannot specify Independent or Simultaneous motion group instructions for circular motion instructions.
- Step** 1 Move the cursor to the line number of any motion statement, except for a circular motion statement. See the following screen for an example.

```

PNS0001                                JOINT 10%
1: L P[1] 1000mm/sec Cnt100
[END]

[ INST ]

```

- 2 Press F1, [INST]. A list of control instructions appears. You will see a screen similar to the following.

```

                                JOINT 10%
1 .....                          5 .....
2 .....                          6 Independent GP
3 .....                          7 Simultaneous GP
4 .....                          8 -- next page --

PNS0001
1: L P[1] 1000mm/sec Cnt100
[END]

[ INST ]

```

- 3 Select Independent GP or Simultaneous GP. The contents of line 1 in the teach pendant program is transferred to the other groups. The position data might be changed. See the following screen for an example.

```

PNS0001                                JOINT 10%
1: Independent GP
: GP1 L P[1] 1000mm/sec Cnt100
: GP3 L P[1] 1000mm/sec Cnt100

[ INST ]

```

- 4 For a motion instruction already within a Independent or Simultaneous motion group instruction, change the motion type, speed, and termination type in the same way as for ordinary motion instructions. Refer to Section 6.3 for more information.

NOTE You cannot **add** motion options to an Independent or Simultaneous motion group instruction once it has been created. All motion options must be added to the Independent or Simultaneous motion group instruction before it is created. However, you can delete motion options from an Independent or Simultaneous motion group instruction.

Cautions and Restrictions

When using the motion group instruction function, observe the following cautions and restrictions:

- If a motion instruction is a circular motion, a motion group instruction cannot be specified.
- You cannot add a circular motion type to an Independent or Simultaneous motion group instruction.
- Within a motion group instruction:
 - The position data format for each group cannot be changed (position <-> position register).
 - The position number for each group cannot be changed.
 - Motion options cannot be added; deletion is possible.
 - A group cannot be deleted or created.
- No search can be made for elements in a motion group instruction.
- You cannot perform simultaneous teaching or deletion for the program element replacement function in a motion group instruction.
- Within a motion group instruction, SHIFT + TOUCHUP cannot be used to correct the position.
- If an incremental instruction is deleted from a motion group instruction, the position data used in the motion group instruction will be uninitialized.

If no motion group instruction has been specified, an ordinary motion instruction is executed. It is assumed all groups are executed simultaneously with the same motion type, time, termination type, and motion options. In this case, other groups are synchronized with the group that has the longest travel time.

6.21 POSITION REGISTER LOOK-AHEAD INSTRUCTIONS

While the robot is executing a program, it reads the lines ahead of the line currently being executed (look-ahead execution). The position register look-ahead execution function enables look-ahead execution for position registers.

The position register look-ahead execution function includes the following program instructions:

- LOCK PREG
- UNLOCK PREG

Refer to Section 10.7 for more detailed information on the position register look-ahead execution function.

LOCK PREG

This instruction locks all position registers. It prevents any change from being made to any position register. See Figure 6-105.

LOCK PREG

Figure 6-105. LOCK PREG Instruction

LOCK PREG

UNLOCK PREG

This instruction unlocks the position registers. See Figure 6-106.

UNLOCK PREG

Figure 6-106. UNLOCK PREG Instruction

UNLOCK PREG

6.22 CONDITION MONITOR INSTRUCTIONS

The condition monitor function monitors the condition of an I/O signal, register value, or alarm status, during teach pendant program execution. As soon as the condition is triggered, the specified teach pendant program is executed and interrupts the current program.

Condition monitor instructions are used to control the monitoring of conditions when a program is running. There are two condition monitor instructions used for program monitoring:

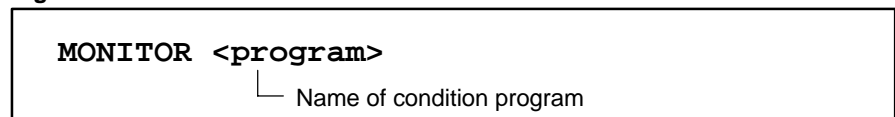
- MONITOR <program>
- MONITOR END <program>
- WHEN <condition> CALL <program>

MONITOR

This instruction starts monitoring the conditions taught in the specified condition program (ch sub type). See Figure 6-107.

MONITOR <program>

Figure 6-107. MONITOR Instruction

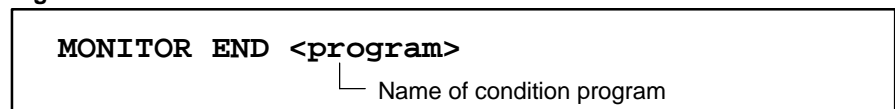


MONITOR END

This instruction stops monitoring the conditions taught in the specified condition program (ch sub type). See Figure 6-108.

MONITOR END <program>

Figure 6-108. MONITOR END Instruction



WHEN <condition> CALL <program>

This instruction defines the conditions for which to monitor. You include WHEN instructions within your condition (ch sub-type) programs. WHEN instructions are the only instructions available when you create condition programs. See Figure 6-109 through Figure 6-111.

In a condition handler program, you can teach multiple WHEN instructions as follows.

```

1: WHEN <cond1> CALL <program1>
2: WHEN <cond2> CALL <program2>
3: WHEN <cond3> CALL <program3>
  
```

You can connect the multiple conditions using AND/OR as follows.

```

1: WHEN <cond1> AND <cond2> CALL <program1>
2: WHEN <cond1> OR <cond2> OR <cond3> CALL
<program2>
  
```

NOTE You cannot use both AND and OR in the same WHEN instruction.

Example

WHEN ... = ... CALL ...

Figure 6-109. Condition for Register, System Variable, and I/O Parameters

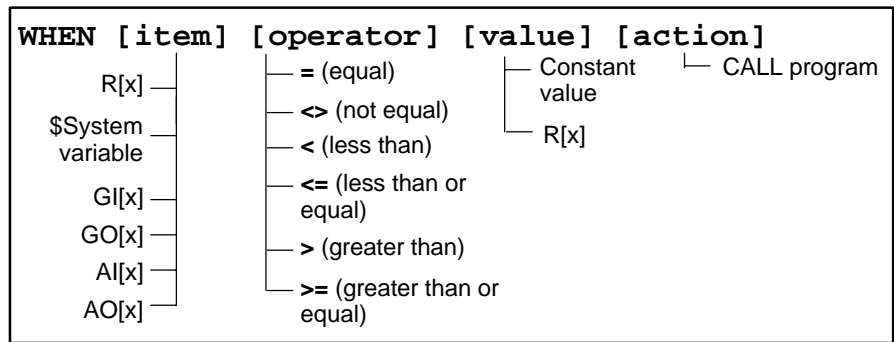
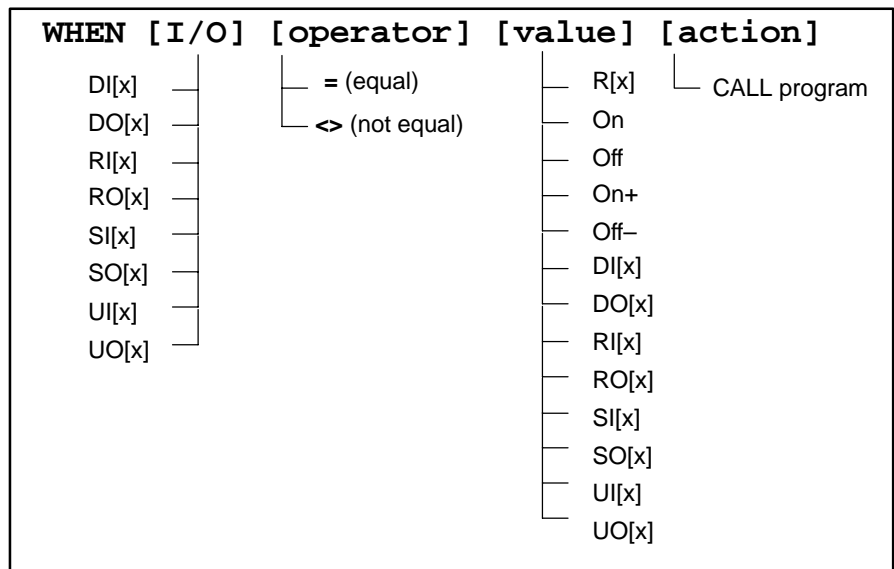


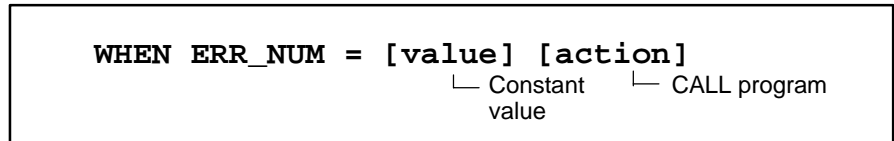
Figure 6-110. Condition2 for I/O



Example

WHEN ... = ... CALL ...

Figure 6-111. Condition for Error Status



ERR_NUM = aaabbb

aaa : Error facility code (decimal); Refer to Section A.1.1.

bbb : Error number (decimal)

Example: WHEN ERR_NUM=11006, CALL PROG_A

This means “SRVO-006 Hand broken” error because the SRVO facility code is 11.

If 0 is specified as error number “aaabbb”, whenever any error occurs, the condition is satisfied.

6.23 PAYLOAD INSTRUCTION

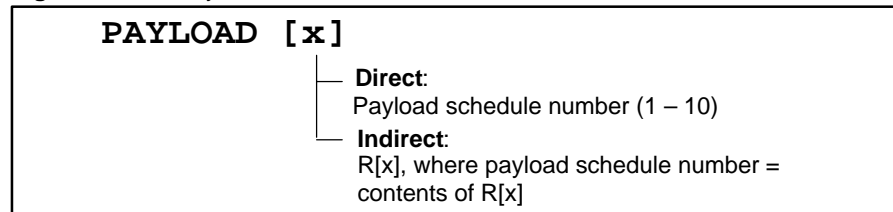
For some applications, you might need to adjust the payload several times within your teach pendant program. For example, if your application requires a change of end-of-arm tooling, you will need to adjust payload information to reflect this change.

See Figure 6-112

PAYLOAD [x]

PAYLOAD [x]

Figure 6-112. Payload Instruction



The PAYLOAD[x] instruction allows you to specify the *payload schedule* to use. You can specify up to 10 different sets of payload information. Each set of payload information corresponds to a schedule number.

High sensitivity collision detection requires the proper setting of payload information. If the payload changes during your application, you must use the PAYLOAD[x] instruction to select the appropriate payload schedule.

Before you use a PAYLOAD[x] instruction, you must make sure you have set up the payload schedule that corresponds to the one you specify. Refer to Section 3.20 for information on setting up payloads.

See Figure 6-113 for an example of using the PAYLOAD[GPx:y] instruction in a teach pendant program.

Figure 6-113. Example of Using PAYLOAD[GPx:y] Instructions in a Teach Pendant Program

```

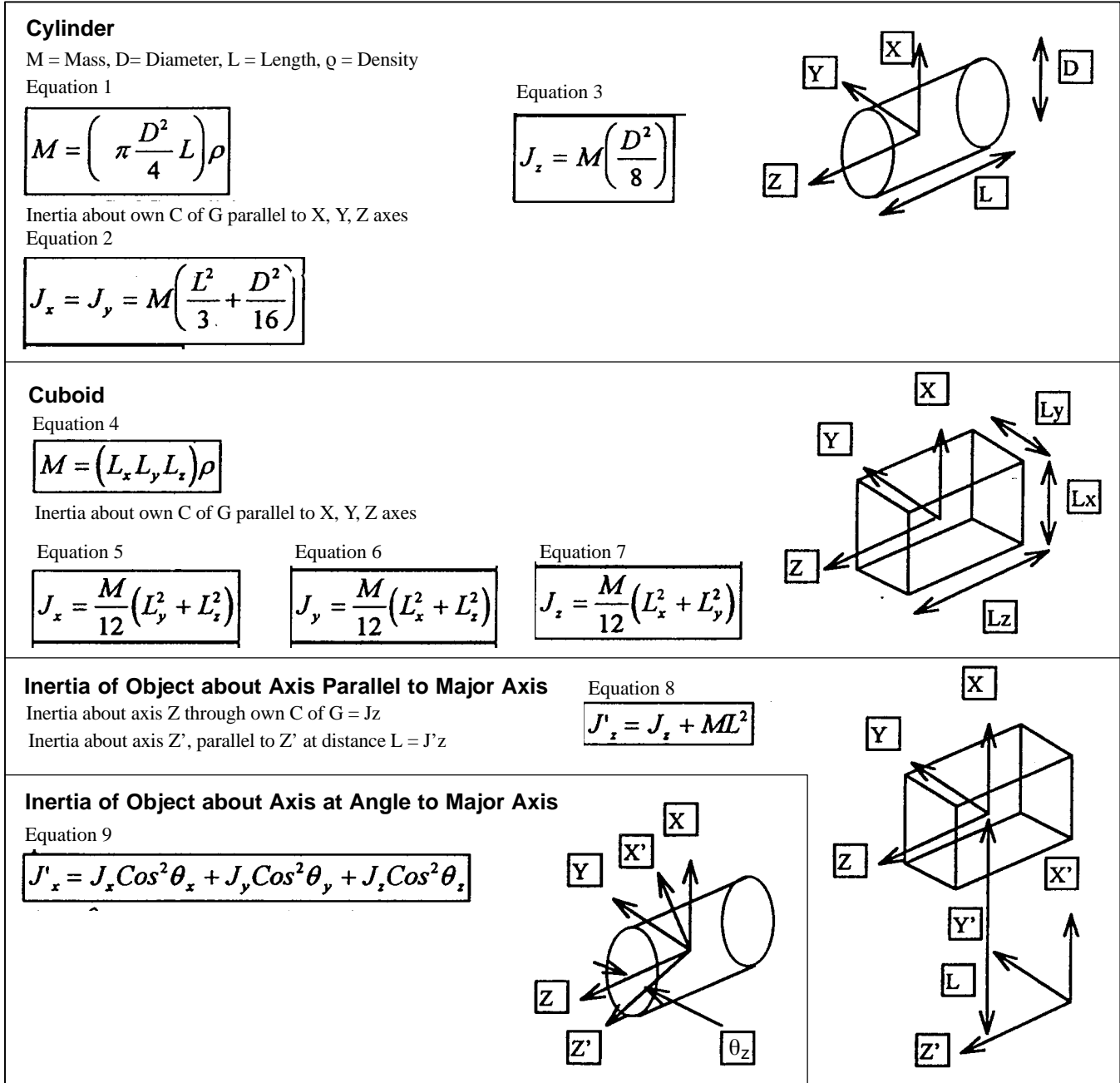
52:  PAYLOAD [GP1:1]
53:  L P[1] 500mm/sec CNT100
54:  L P[2] 2000mm/sec CNT100
55:  L P[3] 500mm/sec FINE
56:  CALL toolchng
57:  PAYLOAD [GP1:2]
58:  L P[2] 500mm/sec CNT100
59:  L P[1] 2000mm/sec/ CNT100
60:  L P[5] 500mm/sec FINE
61:  CALL toolchng
62:  PAYLOAD [GP1:1]
63:  L P[1] 500mm/sec CNT100

```

Inertia Equations

Refer to Figure 6-114 for inertia equations to use in calculating inertia.

Figure 6-114. Inertia Equations



6.24 COLLISION GUARD INSTRUCTIONS

COL DETECT ON COL DETECT OFF

COL DETECT OFF

COL DETECT ON

You can use the Collision Guard instructions to control Collision Guard during programmed motion.

By default, Collision Guard is enabled.

- To **disable** Collision Guard, include the COL DETECT OFF instruction in a teach pendant program.
- To **enable** Collision Guard that has been disabled previously, include the COL DETECT ON instruction in a teach pendant program. Since Collision Guard is always enabled by default, you need to use the COL DETECT ON instruction only if you have previously used the COL DETECT OFF instruction.

See Figure 6-115 for an example of how to use these instructions in a teach pendant program.

Figure 6-115. Example of Enabling and Disabling Collision Guard in a Teach Pendant Program

```
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[5] 50% FINE
```

Refer to Section 10.19 for more information on Collision Guard setup.

7 TESTING A PROGRAM & RUNNING PRODUCTION

7

TESTING A PROGRAM AND RUNNING PRODUCTION

Topics In This Chapter		Page
Program Pause and Recovery	You can EMERGENCY STOP or HOLD any program running in production . . .	7-2
	• EMERGENCY STOP and Recovery	7-2
	• HOLD and Recovery	7-3
	• Setting Tolerance for Resuming a Program	7-4
Test Cycle	You must test your program before you run production	7-9
	• Test Cycle Setup	7-10
	• Single Step Testing	7-12
	• Continuous Testing	7-16
• Monitoring Programs	7-20	
Release Wait	During program execution, release wait allows you to override pauses in the program when the robot is waiting for I/O conditions to be satisfied.	7-21
Production Operation	Production operation is automatic execution of the program	7-22
	• Standard Operator Panel Cycle Start Production	7-22
	• User Operator Panel Start	7-24
	• Robot Service Request (RSR) Production Start	7-25
• Program Number Select (PNS) and UOP Production Start	7-27	
Adjusting Program Information During Production Run	During production or program run, you might need to adjust position information without stopping program execution.	7-29
Maintenance and Repair	You can use macro commands that appear on the MANUAL FCTNS menu to perform maintenance and repair procedures during production	7-33

Testing

Testing includes

- Running a program by
 - **Stepping** through each line of the program
 - **Continuously** running the program for a single cycle
 - Enabling and disabling robot motion, and the palletizing process during testing to verify each instruction of the program
- Monitoring your program
- Controlling inputs and outputs by
 - Forcing outputs
 - Simulating inputs and outputs
- **Adjusting program information** without stopping the program or production.

Running Production

Running production includes

- Running a thoroughly tested program **continuously and repeatedly with all production conditions enabled**
- Performing maintenance procedures when necessary
- Executing multiple programs

NOTE During testing and running production, you also must know how to **stop the program** if there is a safety problem or adjustment to make, and then how to **restart the program**.

7.1 PROGRAM PAUSE AND RECOVERY

You can **EMERGENCY STOP** or **HOLD** any program running in production using

- The **EMERGENCY STOP button** on the teach pendant or operator panel
- The **HOLD button** on the teach pendant or operator panel

7.1.1 EMERGENCY STOP and Recovery

Press the **EMERGENCY STOP** button on the operator panel or teach pendant to stop the robot immediately. Pressing the **EMERGENCY STOP** button

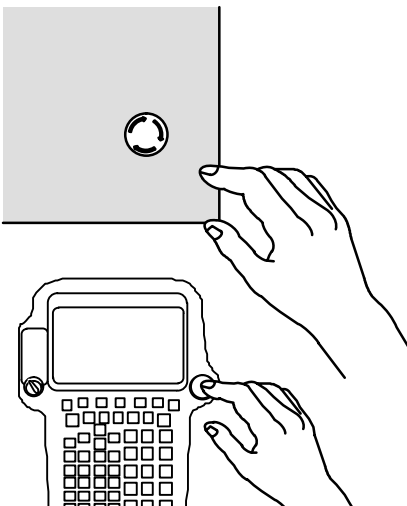
- Stops the running program
- Turns off drive power to the robot servo system
- Applies the brakes to the robot

Use Procedure 7-1 to perform an emergency stop. Use Procedure 7-2 to recover from an emergency stop.

Procedure 7-1 **EMERGENCY STOP**

Step 1 Press the **EMERGENCY STOP** button on the teach pendant or operator panel. This causes an **EMERGENCY STOP** fault.

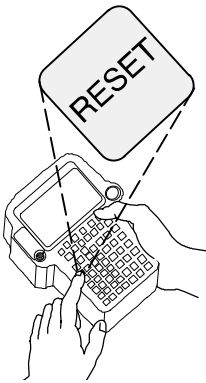
Step 2 Fix the problem that caused you to press the **EMERGENCY STOP** button.



Procedure 7-2 Recovery from EMERGENCY STOP



Step 1 Turn the EMERGENCY STOP button clockwise to release it.



2 Press the RESET button on the teach pendant or operator panel.

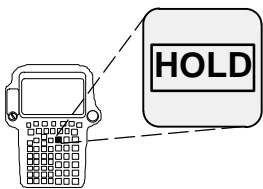
7.1.2 HOLD and Recovery

Press the HOLD button on the teach pendant or operator panel to pause a running program. Pressing the HOLD button

- Pauses a running program
- Causes the robot to decelerate and come to a controlled stop

Use Procedure 7-3 to recover from a held program.

Procedure 7-3 HOLD and Recovery



Step 1 Press the HOLD button on the teach pendant or operator panel.

2 Fix the problem that caused you to press the HOLD button.

3 Press the RESET button on the teach pendant or operator panel to reset the fault.

4 Press CYCLE START to resume program execution.

NOTE Hold is released when program execution begins.

To display help information, press NEXT, >, and then press F1, HELP. When you are finished displaying help information, press PREV.

7.1.3 Setting Tolerance for Resuming a Program

If you have the Control Reliable (RS-1/RS-4) option and resume a program while in AUTO mode, and the distance between the resume position and the stop position is greater than the *stop tolerance*, a prompt box is displayed. When this occurs, you must perform specific actions to restart the program.

Stop tolerance is the amount of distance allowed between the resume robot position and the robot stop position. You can specify the following stop tolerances:

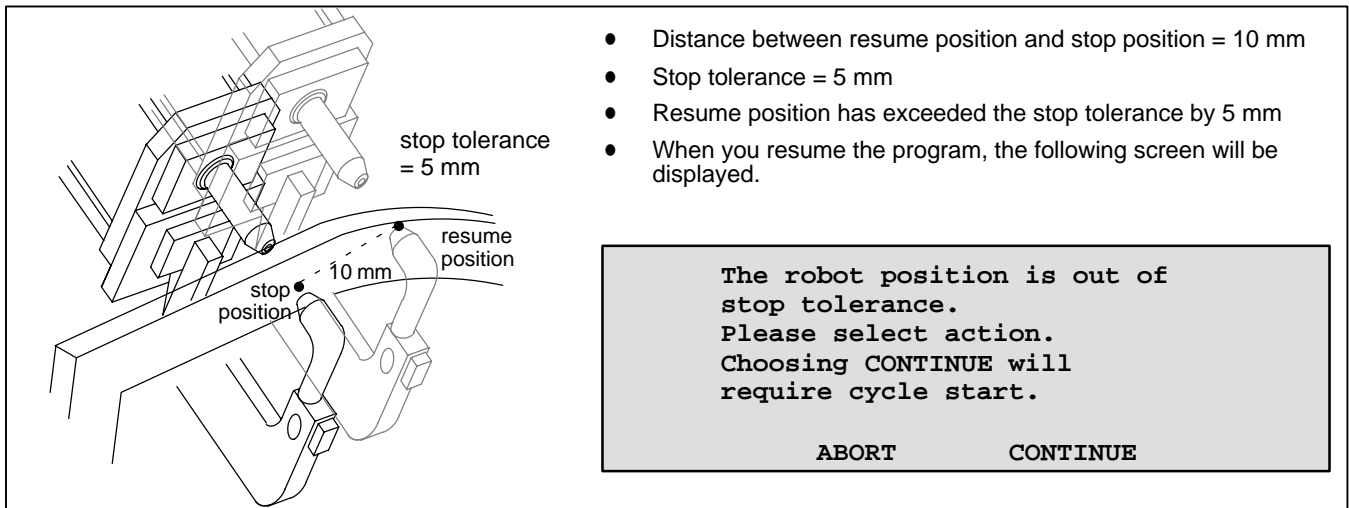
- **Distance tolerance**, for the location components of the position (x,y,z)
- **Orientation tolerance**, for the orientation components of the position (w,p,r)
- **Axes tolerances** for the joint angle (**rotary axes**) or distance location (**translational axes**) of any extended axes, if they are used.

Operation

The following sequence illustrates the operation of the resuming a program for which a tolerance has been set (see Figure 7-1):

1. A running program is paused. The position in which the robot stops is called the **stop position**.
2. The robot is moved to another position prior to resuming the program. This is called the **resume position**.
3. Cycle start is issued to resume the program.
 - If the distance between the stop position and the resume position is **greater than** the stop tolerance
 - a. A prompt box is displayed and the program is not resumed.
 - b. You choose the next action:
 - Abort the program.
 - Restart the program in its current position.
 - Change the mode to T1 or T2 and move the robot to another position by jogging it. Then, restart the program. The current position will be rechecked for tolerance.
 - If the distance between the stop position and the resume position is **less than** the stop tolerance, the program is resumed.

Figure 7-1. Resume Tolerance Example



Limitations

You cannot set tolerance for resume in the following cases:

- In line tracking programs
- In programs that use constant joint path (CJP)
- KAREL motion programs

Setting Up Tolerance for Resuming a Program

You can define the tolerances that will be used when programs run in AUTO mode are paused and then resumed. If you do not define the tolerances, the default values will be used.

Table 7-1 lists the tolerances you can set. Use Procedure 7-4 to set up tolerance for resuming a program.

Table 7-1. Tolerance Setup Items

ITEM	DESCRIPTION
Group default: 1 min: 1 max: 5	This is the motion group number of the axes for which you are setting tolerances. If you have a multiple motion group system, this is a number from 1 to 5. If you do not have a multiple motion group system, make sure Group is set to 1.
Enable Tolerance Checking default: YES	This specifies whether tolerances will be checked during program resume. <ul style="list-style-type: none"> • YES means axes will be checked for tolerance during program resume. • NO means axes will not be checked for tolerance during program resume.
Distance Tolerance default: 250 mm min: 0.1 mm max: 1000 mm	This is the location distance between the resume robot position and the position in which the robot stopped when the program was stopped.
Orientation Tolerance default: 20 degrees min: 0.1 degree max: 80 degrees	This is the angular or orientation distance between the resume robot tool approach vector and the stop position of the tool approach vector.

Table 7-1. (Cont'd) Tolerance Setup Items

ITEM	DESCRIPTION
Axes Tolerance Rotational Axes default: 20 degrees min: 0.1 degree max: 80 degrees	If the robot has extended axes, this is the angular or orientation distance between the resume extended axes position and the corresponding extended axes of the robot stop position.
Axes Tolerance Translational Axes default: 250 mm min: 0.1 mm max: 1000 mm	If the robot has extended axes, this is the location distance between the resume extended axes position and the corresponding extended axes of the robot stop position.

Procedure 7-4 Setting Up Tolerance for Resuming a Program

- Step**
- 1 Press **MENUS**.
 - 2 Select **SETUP**.
 - 3 Press **F1**, [**TYPE**].
 - 4 Select **RESUME TOL.** You will see a screen similar to the following.

SETUP RESUME TOL.		JOINT	50%
		1/6	
1	Group:	1	
2	Enable tolerance checking:	YES	
3	Distance Tolerance (mm)	250.0	mm
4	Orientation Tolerance (deg)	20.0	deg
Axes Tolerance			
5	Rotational axes (deg)	20.0	deg
6	Translational axes (mm)	250.0	mm
[TYPE]			

- 5 Select each item and set as desired.

Resuming a Program that Exceeds the Stop Tolerance

If you resume a program in which the current robot position exceeds the stop tolerance, you must perform specific procedures appropriate to the conditions. Use Procedure 7-5 to resume a program that exceeds the stop tolerance.

Procedure 7-5 Resuming a Program that Exceeds the Stop Tolerance

- Condition**
- The robot is in AUTO mode.
 - The program has been paused. You will see a screen similar to the following.

```

1. Jog to position where TP was enabled
2. Abort program
3. Continue from current position

```

[CHOICE]

- Step**
- 1 Select the appropriate action:
 - To **abort**, go to Step 2.
 - To **continue**, go to Step 3.
 - To **move the robot back into tolerance and restart**, go to Step 4.

Abort the Program

- 2 To **abort** the program, move the cursor to ABORT and press ENTER or press F4, [CHOICE], and select Abort program.
The program is aborted.

Continue the Program from the Current Position

- 3 To **continue** the program from the current position
 - a Move the cursor to CONTINUE and press ENTER or press F4, [CHOICE], and select Continue from current position.
The program is still paused.

- b Input the start signal again to restart the program at its present position.

The robot will move from the current position to the stop position and continue the program. The robot does not check whether it is out of tolerance, and the prompt box is not displayed again.

NOTE If the restarted program is the paused program, the appropriate recovery functions are executed. Refer to Section 10.20 for more information.

**Move Robot into Tolerance
and Restart**

- 4 To move the robot into tolerance and restart the program,**
- a** Set the MODE SELECT switch to the T1 or T2 position.
 - b** Jog the robot to a position that is within the tolerance.
 - c** Set the MODE SELECT switch to the AUTO position.
 - d** Input the start signal again to restart the program at its present position.

The robot will check again whether it is out of tolerance. If it is out of tolerance, the prompt box is displayed again. Repeat Steps 4a through 4d until the robot is within the resume tolerance.

NOTE If the restarted program is the paused program, the appropriate recovery functions are executed. Refer to Section 10.20 for more information.

7.2 TEST CYCLE

You must test your program before you run production. For each step in the following test plan you must set up test cycle conditions as appropriate for the kind of testing you are performing.

A typical testing plan will

1. Single step through the program using the teach pendant while the process is disabled to check the robot motion, other instructions, and I/O.
2. Continuously run the program using the teach pendant at a low speed with the process disabled.
3. Continuously run the program using the operator panel at high speed with the process disabled to check the robot positions and timing.
4. Continuously run the program using the operator panel with the process enabled to verify the process.

7.2.1 Test Cycle Setup

Setting up the test cycle allows you to control the conditions for test running a program. These conditions are in effect any time a program is run until you decide to change the conditions.

You can set the test cycle conditions listed and described in Table 7-2. Use Procedure 7-6 to set up test cycle conditions.

Table 7-2. Test Cycle Conditions

TEST CYCLE CONDITION	DESCRIPTION
Machine Lock	Determines whether the robot and any extended axes will move during the test cycle. If set to OFF , the robot will move . If set to ON , the robot will not move .
Dry Run	Determines whether the EOAT will function during the test cycle. If set to OFF , the EOAT will function . If set to ON , the EOAT will not function .
Cartesian Dry Run Speed	Determines the speed at which the robot will move when using Cartesian motion (linear or circular moves) if dry run is set to ON.
Joint Dry Run Speed	Determines the speed at which the robot will move when using joint motion if dry run is set to ON. A joint dry run speed of 100 will test run the program at 100% of the programmed speed.
Digital/Analog	Determines whether digital/analog input and output signals will turn on and off during the test cycle. If set to ENABLED , the input and output signals will turn on and off . If set to DISABLED , the input and output signals will not function.
Step Statement Type	Allows you to select at which statements the robot will pause between steps. There are five statement types: <ul style="list-style-type: none"> ● STATEMENT – The program pauses when the execution of each program statement is completed. The program pauses after executing each step in a routine. ● MOTION – The program pauses when the execution of each motion statement is completed. ● ROUTINE – The program pauses after each statement is executed, but each routine is executed as one continuous step. ● TP & MOTION – The program pauses when the execution of a teach pendant statement or when a KAREL motion statement is completed.
Step Path Node	Specifies whether to pause the program after each path node during a KAREL MOTION ALONG statement. When set to ON, the program will pause after each path node. When set to OFF, each path will be executed as one continuous step.

Procedure 7-6 Setting Up Test Cycle Conditions

- Step**
- 1 Press SELECT.
 - 2 Select the program you want to test and press ENTER.
 - 3 Press MENUS.
 - 4 Select 2, TEST CYCLE. You will see a screen similar to the following.

TEST CYCLE		JOINT	50%
			1/7
GROUP: 1			
1	Robot lock:	OFF	
2	Dry run:	OFF	
3	Cart.dry run speed:	300.000 mm/s	
4	Joint dry run speed:	25.000 %	
5	Digital/Analog I/O:	DISABLE	
6	Step statement type:	STATEMENT	
7	Step path node:	OFF	
[TYPE]	GROUP	ON	OFF

- 5 Set Test Cycle conditions as desired.

7.2.2 Single Step Testing

Single step testing is running individual program instructions one at a time. You use the teach pendant to single step the current program displayed on the teach pendant screen.

If you have the Control Reliable (RS-1/RS-4) option, the MODE SELECT switch must be in the T1 or T2 position to test a program in single steps using the teach pendant. If you test a program in T1 mode, the robot speed will be no greater than 250mm/sec, regardless of any other speed settings. Refer to Section 1.2.2 for more information on the MODE SELECT switch.

If you have a European controller, refer to “Mode select switch” in 1.2.2 Standard Operator Panel (SOP) for more information about using the mode selection switch.

If you set the singularity stop system variable, \$PARAM_GROUP[n].\$T1T2_SNGSTP, to TRUE, the robot will stop at singularity points while in T1 or T2 mode.

Single step testing can be done two ways:

- Forward
- Backward

Forward

Forward

- Executes one instruction when the SHIFT and FWD keys are pressed and FWD is released.
- Stops when the step is completed or SHIFT is released.
- Executes subprograms one step at a time.

Backward

Backward

- Executes the previous instruction when the SHIFT and BWD keys are pressed and then BWD is released.
- Steps backward from a sub-program to the main program. Before you can do this, you must step forward from the main program to the sub-program. When you return to the main program from the sub program, the cursor pauses on the CALL instruction in the sub program.
- Stops when the step is completed or the SHIFT key is released.
- Can only be done for motion instructions.

NOTE You cannot use backward execution to call the sub program from the main program.

Use Procedure 7-7 to single step test a program.

Figure 7-2 contains an example program which shows how to perform backward execution from the 4th line of the sub program SUB_PROG.

Figure 7-2. Example Program Showing Backward Execution

```

MAIN_PROG
1:
2: R[1] = R[1] + 1
3: J P[1] 100% FINE
4:
5: CALL SUB_PROG
6:
[END]

SUB_PROG
1: SDO[1] = ON
2: SDO[2] = ON
3: L P[2] 1000mm/sec FINE
4: L P[3] 1000mm/sec FINE
[END]

```

1. Start to do the backward execution from 4th line of the SUB_PROG.
Motion: The robot moves from P[3] to P[2].
Cursor: The cursor is on the 3rd line of the SUB_PROG.
2. Do the backward execution again.
Motion: No motion.
Cursor: The cursor is on the 5th line of the MAIN_PROG.
3. Do the backward execution again.
Motion: The robot moves from P[2] to P[1].
Cursor: The cursor is on the 3rd line of the MAIN_PROG.

DISABLE FWD/BWD

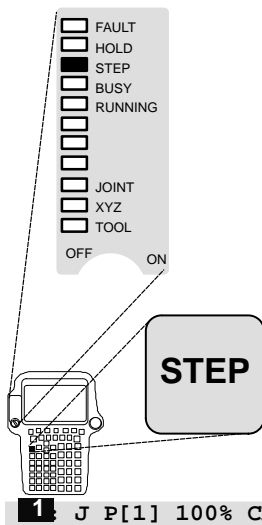
DISABLE FWD/BWD allows you to disable the ability to execute program instructions when the SHIFT and FWD keys or SHIFT and BWD keys are pressed. To use DISABLE FWD/BWD, press FCTN and then select DISABLE FWD/BWD. The ability to use SHIFT FWD and SHIFT BWD will be disabled until you press FCTN and select DISABLE FWD/BWD again.

When the teach pendant FWD and BWD keys are disabled and the teach pendant is enabled, "FBD" is displayed in the upper left hand corner of the teach pendant screen to indicate that you cannot use the teach pendant to run the program.

Procedure 7-7 Single Step Testing

NOTE If you have the Control Reliable (RS-1/RS-4) option, if the MODE SELECT switch is in the T1 position, the robot speed will be no greater than 250mm/sec, regardless of any other speed settings.

NOTE If you have the Control Reliable (RS-1/RS-4) option, you can perform single step testing from the teach pendant only with the MODE SELECT switch in the T1 or T2 position.



Condition

- A program has been created and positions have been recorded.
- Test cycle conditions have been set. (Procedure 7-6)
- All personnel and unnecessary equipment are out of the workcell.
- If you have the Control Reliable (RS-1/RS-4) option, the MODE SELECT switch is in the T1 or T2 position.

Step

- 1 Press SELECT.
- 2 Select the program you want to test and press ENTER.
- 3 Press STEP to enable single step testing. The STEP indicator will turn on.
- 4 Move the cursor to the first line of the program you want to test. The program will start at the current cursor position. You will see a screen similar to the following.

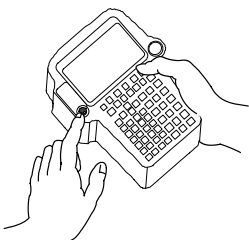
```

Edit Test1234          LINE:1          READY
PRGWELD                WORLD  25%
                        1/9
1  J P[1] 100% CNT50
2: PALLET[1 :pallet_sample]
3: J P[1:A_1] 70% CNT100
4: J P[1:A_2] 50% CNT50
5: L P[1:A_3] 500mm/s FINE
6: L P[1:BTM] 300mm/s FINE
7:   CLOSE HAND [1]
8:   WAIT SDI[1]
9: If PL[3]=[5,*,*] JUMP[11]
[End]
POINT                  TOUCHUP >

```

- 5 Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

NOTE If you have the Control Reliable (RS-1/RS-4) option and you compress the DEADMAN switch fully, robot motion will not be allowed and an error occurs. This is the same as when the DEADMAN switch is released. To clear the error, press the DEADMAN switch in the center position and press RESET.



- 6 Set the speed to the value you want. A low speed is recommended.
- 7 Check program status on the top line of the teach pendant screen. If it is PAUSED, press FCTN and select ABORT (ALL).

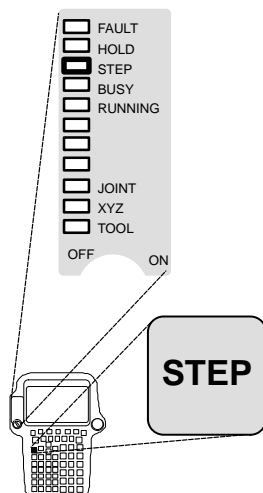
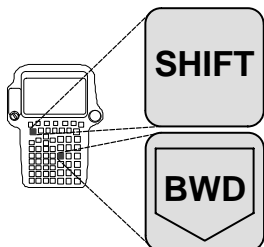
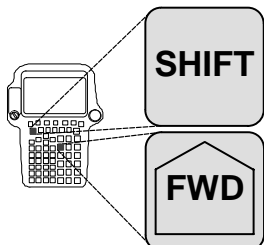
⚠ WARNING

The next step causes a program instruction to run. This could cause the robot to move and other unexpected events to occur. Make sure all personnel and unnecessary equipment are out of the workcell and that all safeguards are in place; otherwise, personnel could be injured and equipment damaged.

In the next step of this procedure, if you want to stop the program instruction before the instruction has finished executing, release the SHIFT key, release the DEADMAN switch, or press the EMERGENCY STOP button.

⚠ WARNING

If you execute motion instructions that contain the remote TCP (RTCP) motion option and you skip motion instructions during testing, the robot might have to change orientation dramatically to reach the destination position. This will cause it to move in a large area. Be aware that this might happen before you skip motion instructions during testing; otherwise, you could injure personnel or damage equipment.



- 8 Test a program instruction.
 - To execute an instruction in the forward direction, press and hold in the SHIFT key and press and release the FWD key. You must hold in the SHIFT key continuously until the instruction has finished executing.
 - To execute an instruction in the backward direction, press and hold in the SHIFT key and press and release the BWD key. You must hold in the SHIFT key continuously until the instruction has completed executing.
- 9 Repeat Step 8 for as many instructions as you want to test.
- 10 Press STEP to disable single step testing. The STEP indicator will turn off.
- 11 Turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.

7.2.3 Continuous Testing

Continuous testing is running a program from beginning to end without stopping. You can test a program continuously using the teach pendant or the operator panel CYCLE START button.

If you have the Control Reliable (RS-1/RS-4) option, to test a program continuously using the teach pendant the MODE SELECT switch must be in the T1 or T2 position. To test a program continuously using the CYCLE START button on the operator panel, the MODE SELECT switch must be in the AUTO position. Refer to Section 1.2.2 for more information on the MODE SELECT switch.

For European controllers, to test a program using the CYCLE START button on the operator panel, the mode select switch is set to AUTO when this switch is installed. Refer to Section 1.2.2, Standard Operator Panel (SOP), for more information on the mode select switch

If you set the singularity stop system variable, \$PARAM_GROUP[n].\$T1T2_SNGSTP, to TRUE, the robot will stop at singularity points while in T1 or T2 mode.

Use Procedure 7-8 to test continuously using the teach pendant. Use Procedure 7-9 to test a program continuously using the operator panel CYCLE START button.

Procedure 7-8 Continuous Testing Using the Teach Pendant

NOTE If you have the Control Reliable (RS-1/RS-4) option, if the MODE SELECT switch is in the T1 position, the robot speed will be no greater than 250mm/sec, regardless of any other speed settings.

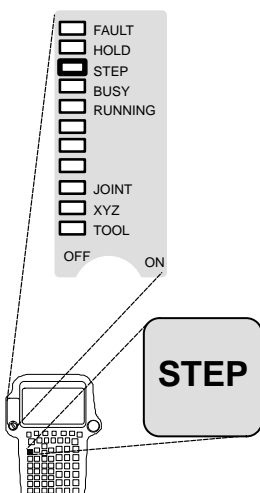
NOTE If you have the Control Reliable (RS-1/RS-4) option, you can perform continuous testing from the teach pendant only with the MODE SELECT switch in the T1 or T2 position.

Condition

- A program has been created and positions have been recorded.
- Test cycle conditions have been set. (Procedure 7-6)
- All personnel and unnecessary equipment are out of the workcell.
- You have tested the program in single step. (Procedure 7-7)
- If you have the Control Reliable (RS-1/RS-4) option, the MODE SELECT switch is in the T1 or T2 position.

Step

- 1 Press SELECT.
- 2 Select the program you want to test and press ENTER.
- 3 Disable single step testing. If the STEP indicator is ON, press STEP to disable it.



1 J P[1] 100% CNT50

- 4 Move the cursor to line 1. The program will start at the current cursor position.
- 5 Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

NOTE If you have the **Control Reliable (RS-1/RS-4)** option and you compress the DEADMAN switch fully, robot motion will not be allowed and an error occurs. This is the same as when the DEADMAN switch is released. To clear the error, press the DEADMAN switch in the center position and press RESET.

- 6 Set the speed to the value you want. First run the speed at 5% – 10% value.
- 7 Check program status on the top line of the teach pendant screen. If it is PAUSED, press FCTN and select ABORT (ALL).

⚠ WARNING

The next step causes a program to run. This could cause the robot to move, the process to run, and other unexpected events to occur. Make sure all personnel and unnecessary equipment are out of the workcell and that all safeguards are in place; otherwise, personnel could be injured and equipment damaged.

In the next step of this procedure, if you want to stop the program instruction before the instruction has finished executing, release the SHIFT key, release the DEADMAN switch, or press the EMERGENCY STOP button.

⚠ WARNING

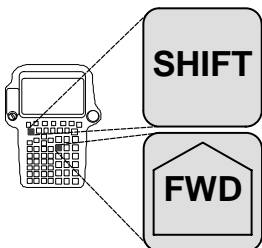
If you execute motion instructions that contain the remote TCP (RTCP) motion option and skip motion instructions during testing, the robot might have to change orientation dramatically to reach the destination position, causing it to move in a large area. Be aware that this might happen before you skip motion instructions during testing; otherwise, you could injure personnel or damage equipment.

NOTE You can test a program continuously in the forward direction only.

NOTE If you have set the singularity stop system variable, \$PARAM_GROUP[n].\$T1T2_SNGSTP, to TRUE, the robot will stop at singularity points while in T1 mode.

- 8 Press and hold down the SHIFT key and press and release the FWD key. You must hold in the SHIFT key continuously until the instruction has finished executing.

Run program in 5% – 10% intervals, up to 100%. If continuous positions are changed, restart the process at a low speed.



Procedure 7-9 Continuous Testing Using the Operator Panel CYCLE START Button

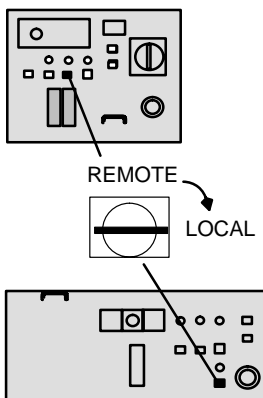
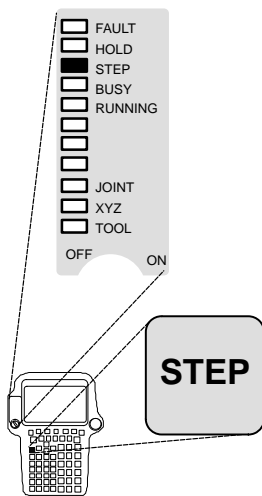
NOTE If you have the Control Reliable (RS-1/RS-4) option, you can perform continuous testing using the CYCLE START button only with the MODE SELECT switch in the AUTO position.

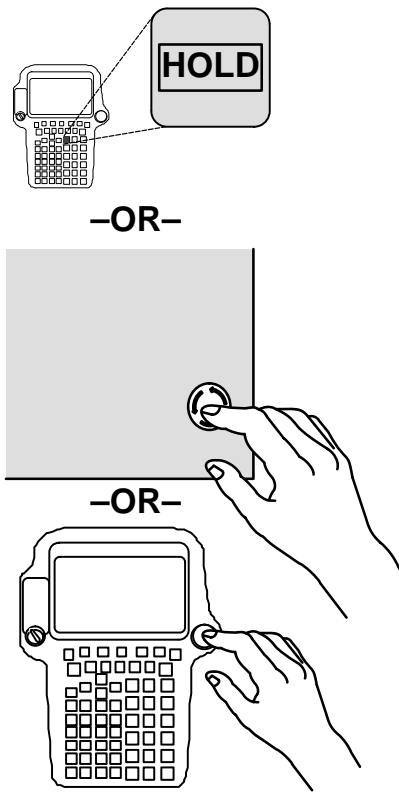
Condition

- The program has been created and positions recorded.
- Test cycle conditions have been set.(Procedure 7-6)
- All personnel and unnecessary equipment are out of the workcell.
- You have tested the program in both single step (Procedure 7-7) and continuous (Procedure 7-8) using the teach pendant.
- If you have a European controller, the mode select switch is set to AUTO when this switch is installed.
- If you have the Control Reliable (RS-1/RS-4) option, the MODE SELECT switch is in the AUTO position.

Step

- 1 Press SELECT.
 - 2 Disable single step testing. If the STEP indicator is on, press STEP to turn it off.
 - 3 Select the program you want to test and press ENTER.
 - 4 Set the speed to the value you want.
- When CYCLE START is used, the speed override might automatically be set to 100%.
- 5 Turn the REMOTE/LOCAL switch on the standard operator panel to LOCAL.



**WARNING**

The next step causes a program instruction to run. This could cause the robot to move, the process to run, and other unexpected events to occur. Make sure all personnel and unnecessary equipment are out of the workcell and that all safeguards are in place; otherwise, personnel could be injured and equipment damaged.

If you want to stop the program before it has finished executing, press the HOLD button for a decelerated stop, or the EMERGENCY STOP button for an immediate stop.

⚠ WARNING

If you execute motion instructions that contain the remote TCP (RTCP) motion option and skip motion instructions during testing, the robot might have to change orientation dramatically to reach the destination position, causing it to move in a large area. Be aware that this might happen before you skip motion instructions during testing; otherwise, you could injure personnel or damage equipment.

- 6 Press the CYCLE START button on the operator panel.

You will be prompted to confirm program execution. If you select YES, then you must press CYCLE START again to run the program.

7.2.4 Monitoring Programs

You can monitor a running program from the SELECT menu. When you monitor a running program, the program is displayed and the cursor highlights the line currently being executed.

Use Procedure 7-10 to monitor a running program.

Procedure 7-10 Monitoring a Running Program

Condition ■ The program you want to monitor is currently executing.

Step 1 Press SELECT.

Select		JOINT 10%	
No.	Program name	50983 BYTES FREE	1/6
		Comment	
1	SUB1	[]
2	MAIN25	[]
3	PRG7	[]
4			
5			
6			
[TYPE] CREATE DELETE MONITOR [ATTR] >			
COPY DETAIL LOAD SAVE PRINT >			

2 Press F4, MONITOR.

The program will be displayed on the screen. The cursor will be on the line number of the instruction that is currently being executed. The cursor will move to each instruction as it is executed.

3 To look at another area of the program while the program is being displayed, press F2, LOOK. When you want the cursor to return to the line number of the instruction currently being executed, press F2, MONITOR.

7.3 RELEASE WAIT

During program execution, release wait allows you to override pauses in the program when the robot is waiting for I/O conditions to be satisfied. Release wait works only when a program is running.

**WARNING**

Be careful when using release wait. Overriding I/O or wait periods can cause the robot to move or equipment to operate unexpectedly.

Procedure 7-11 Using Release Wait

- Condition**
- A program is running.
 - The running program is waiting for I/O conditions to be satisfied.

- Step**
- 1 Press the FCTN key.

**WARNING**

Be careful when using release wait. Overriding I/O or wait periods can cause the robot to move or equipment to operate unexpectedly.

- 2 Select RELEASE WAIT.
 - **If an active wait is pending**, the program will pause. Resume the program when you are ready, using the method you used to run the program. For safety reasons, the program will pause again at the next application instruction.
 - **If no active wait is pending**, nothing will happen.

7.4 PRODUCTION OPERATION

Production operation is automatic execution of the program. The program runs continuously and repeatedly with full speed, palletizing, I/O, and motion conditions enabled.

If you have the Control Reliable (RS-1/RS-4) option, the MODE SELECT switch must be in the AUTO position to perform production operation. Refer to Section 1.2.2 for more information on the MODE SELECT switch.

If you set the singularity stop system variable, \$PARAM_GROUP[n].\$AUTO_SNGSTP, to FALSE, the robot will pass through singularity points while in AUTO mode.

There are four ways to run production:

- SOP (Standard Operator Panel) CYCLE START
- UOP (User Operator Panel) START
- Robot Service Request (RSR)
- Program Number Select (PNS) and UOP PRODUCTION START

7.4.1 Standard Operator Panel Cycle Start Production

A Standard Operator Panel Cycle Start is a method of automatically running the selected program. Selecting Cycle Start implies that you are not using RSR or PNS, but will use the CYCLE START input on the Standard Operator Panel (SOP) to initiate production operation.

Refer to Section 3.3 for more information about setting up to run production using an SOP cycle start.

The CYCLE START input will run the selected program for one cycle unless the program contains a loop, or your system is set up to send the CYCLE START input again as soon as the controller determines that the program has finished a cycle.

Refer to Chapter 2 for more information about setting up to run production using SOP cycle start.

If you have the Control Reliable (RS-1/RS-4) option, the MODE SELECT switch must be in the AUTO position to perform SOP CYCLE START. Refer to Section 1.2.2 for more information on the MODE SELECT switch.

If you have a European controller, refer to Section 1.2.2 for more information on MODE SELECT SWITCH.

If you set the singularity stop system variable, \$PARAM_GROUP[n].\$AUTO_SNGSTP to FALSE, the robot will pass through singularity points while in AUTO mode. If you change the value of this variable, you must cycle power for the change to take effect.

Use Procedure 7-12 to run production using Standard Operator Panel (SOP) Cycle Start.

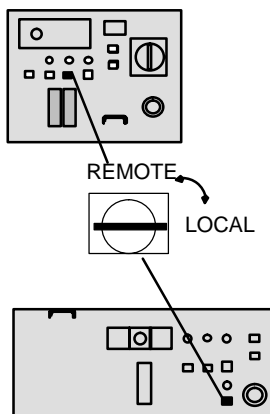
Procedure 7-12 Running Production Using Standard Operator Panel (SOP) Cycle Start

- Condition**
- The robot is powered up and all faults have been corrected.
 - The program has been tested thoroughly and found to operate correctly.
 - All personnel and unnecessary equipment are out of the workcell.
 - All safeguards have been installed and are functioning correctly.
 - Any other conditions related to the application or robot have been satisfied.
 - Test cycle conditions are set properly to allow robot motion, palletizing, I/O, and full production speed.
 - Single step testing is disabled and the STEP LED is not illuminated.
 - If you have the Control Reliable (RS-1/RS-4) option, the MODE SELECT switch is in the AUTO position.
 - If you have a European controller, the MODE SELECT switch is set to AUTO when this switch is installed.



WARNING

This procedure starts production run. Make sure all safety barriers are in place, all personnel are outside of the workcell, all equipment is in place, and all production conditions have been met before you continue, otherwise, personnel can be injured and equipment damaged.



- Step**
- 1 Set the LOCAL/REMOTE keyswitch on the operator panel to LOCAL.
 - 2 Select the program using the SELECT menu.
 - 3 Press the CYCLE START button on the standard operator panel.

7.4.2 User Operator Panel Start

A User Operator Panel Start is a method of automatically running the selected program. Selecting Start implies that you are not using RSR or PNS, but will use the START input on the User Operator Panel (UOP) to initiate production operation.

The START input will run the selected program for one cycle unless the program contains a loop, or your system is set up to send the START input again as soon as the controller determines that the program has finished a cycle.

Refer to Chapter 2 for more information about setting up to run production using UOP cycle start.

Remote conditions are satisfied before running the selected program. Refer to 3.3.2 UOP Output Signals for more information about remote conditions.

If you have the Control Reliable (RS-1/RS-4) option, the MODE SELECT switch must be in the AUTO position to perform DIN CYCLE START. Refer to Section 1.2.2 for more information on the MODE SELECT switch.

If you have a European controller, refer to Section 1.2.2 for more information on mode select switch.

If you set the singularity stop system variable, \$PARAM_GROUP[n].\$AUTO_SNGSTP, to FALSE, the robot will pass through singularity points while in AUTO mode.

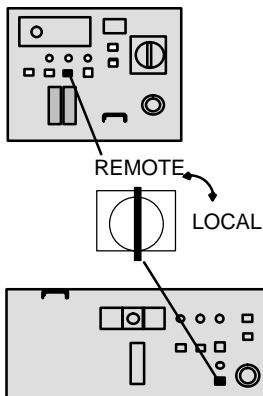
Procedure 7-13 Running Production Using User Operator Panel (UOP) Start

- | | |
|------------------|---|
| Condition | <ul style="list-style-type: none"> ■ The robot is powered up and all faults have been corrected. ■ The program has been tested thoroughly and found to operate correctly. ■ All personnel and unnecessary equipment are out of the workcell. ■ All safeguards have been installed and are functioning correctly. ■ Any other conditions related to the application or robot have been satisfied. ■ UOP has been correctly installed and configured. ■ The UOP UI enable signal *ENBL is ON. ■ The UOP UI safety fence digital signal *SFSPD is ON. ■ Test cycle conditions are set properly to allow robot motion, palletizing, welding, I/O, and full production speed. ■ Single step testing is disabled and the STEP LED is not illuminated. |
|------------------|---|

- RSR and PNS are disabled.
- If you have the Control Reliable (RS-1/RS-4) option, the MODE SELECT switch is in the AUTO position.
- If you have a European controller, the mode select switch is set to AUTO when this switch is installed.

**WARNING**

This procedure starts production run. Make sure all safety barriers are in place, all personnel are outside of the workcell, all equipment is in place, and all production conditions have been met before you continue; otherwise, personnel can be injured and equipment damaged.

**Step**

- 1 Set the LOCAL/REMOTE keyswitch on the operator panel to REMOTE.
- 2 Select the program using the SELECT menu.
- 3 Press the UOP START button to start the application program.

7.4.3 Robot Service Request (RSR) Production Start

A robot service request (RSR) is a request for service from an external device. That request comes from a digital input signal on a preassigned RSR input line.

You can use up to four robot service request signals: RSR1, RSR2, RSR3, and RSR4.

Refer to Chapter 2 for more information about setting up to run production using RSR.

Remote conditions are satisfied before RSR production start.

Refer to 3.3.2 UOP Output Signals for more information about remote conditions.

If you have the Control Reliable (RS-1/RS-4) option, the MODE SELECT switch must be in the AUTO position to perform RSR Production Start. Refer to Section 1.2.2 for more information on the MODE SELECT switch.

If you have a European controller, refer to Section 1.2.2, Standard Operator Panel (SOP), for more information on the mode select switch.

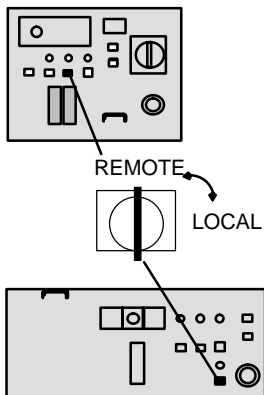
If you set the singularity stop system variable, `$PARAM_GROUP[n].$AUTO_SNGSTP`, to FALSE, the robot will pass through singularity points while in AUTO mode.

Procedure 7-14 Running Production Using Robot Service Requests (RSR)

- Condition**
- The robot is powered up and all faults have been corrected and cleared.
 - The program has been tested thoroughly and found to operate correctly.
 - All personnel and unnecessary equipment are out of the workcell.
 - All safeguards have been installed and are functioning correctly.
 - Any other conditions related to the application or robot have been satisfied.
 - UOP has been correctly installed and configured.
 - The UOP UI enable signal *ENBL is ON.
 - The UOP UI safety fence digital signal *SFSPD is ON.
 - Test cycle conditions are set properly to allow robot motion, palletizing, I/O, and full production speed.
 - Single step testing is disabled and the STEP LED is not illuminated.
 - RSR setup has been completed.
 - PNS is disabled.
 - If you have the Control Reliable (RS-1/RS-4) option, the MODE SELECT switch is in the AUTO position.
 - If you have a European controller, the mode select switch is set to AUTO when this switch is installed.

**WARNING**

This procedure starts production run. Make sure all safety barriers are in place, all personnel are outside of the workcell, all equipment is in place, and all production conditions have been met before you continue, otherwise, personnel can be injured and equipment damaged.

**Step**

- 1 Set the LOCAL/REMOTE keyswitch on the operator panel to REMOTE. When the RSR input is received, production operation begins as long as all UOP UI conditions are satisfied.

7.4.4 Program Number Select (PNS) and UOP Production Start

A program number select (PNS) is a method of selecting a program to be run by some external device. The name of the program to be run is received by the controller as a group of input signals from an external device on a total of eight PNS input lines.

Refer to Chapter 3, Section 3.9.2 for more information about setting up to run production using PNS.

Remote conditions are satisfied before PNS production start. Refer to 3.3.2 UOP Output Signals for more information about remote conditions.

If you have the Control Reliable (RS-1/RS-4) option, the MODE SELECT switch must be in the AUTO position to perform PNS Production Start. Refer to Section 1.2.2 for more information on the MODE SELECT switch.

If you have a European controller, refer to Section 1.2.2, Standard Operator Panel (SOP), for more information on the mode select switch.

If you set the singularity stop system variable, \$PARAM_GROUP[n].\$AUTO_SNGSTP, to FALSE, the robot will pass through singularity points while in AUTO mode.

Procedure 7-15 Running Production Using Program Number Select (PNS) and UOP Production Start

- | | |
|------------------|---|
| Condition | <ul style="list-style-type: none"> ■ The robot is powered up and all faults have been corrected. ■ The program has been tested thoroughly and found to operate correctly. ■ All personnel and unnecessary equipment are out of the workcell. ■ All safeguards have been installed and are functioning correctly. ■ Any other conditions related to the application or robot have been satisfied. ■ UOP has been correctly installed and configured. ■ The UOP UI enable signal *ENBL is ON. ■ The UOP UI safety fence digital signal *SFSPD is ON. ■ Test cycle conditions are set properly to allow robot motion, palletizing, I/O, and full production speed. ■ Single step testing is disabled and the STEP LED is not illuminated. ■ PNS setup has been completed. Refer to Section 3.9.2. ■ RSR is disabled. ■ If you have the Control Reliable (RS-1/RS-4) option, the MODE SELECT switch is in the AUTO position. |
|------------------|---|

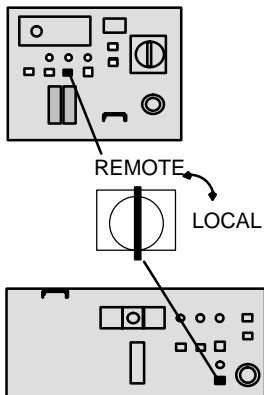
- If you have a European controller, the mode select switch is set to AUTO when it is installed.

**WARNING**

This procedure starts production run. Make sure all safety barriers are in place, all personnel are outside of the workcell, all equipment is in place, and all production conditions have been met before you continue, otherwise, personnel can be injured and equipment damaged.

**WARNING**

Failure to follow this procedure results in the filling of the temporary memory in the R-J2 controller causing the process CPU to be locked into a busy and running condition. This could cause injury to personnel and damage to equipment. Make sure your PLC logic is correct and does not contain a high rate of production start calls.

**Step**

- 1 Set the LOCAL/REMOTE keyswitch on the operator panel to REMOTE.
- 2 Set the 8 bit PNS input to the number that when added to the base number will determine which program is selected. Refer to Chapter 3, Section 3.9.2 for more information about PNS.
- 3 Strobe the PNSTROBE input. When the controller receives the input signal, the selected program will be displayed on the teach pendant screen. The ACK UOP signal indicates what binary input is being received. This stays ON until a new program is selected.
- 4 Press the production start button on the user operator panel to start production operation or, if your system uses a PLC, production operations will begin as soon as the PROD_START input is received.

Refer to Section 3.3 for more information about setting up to run production using a UOP START.

7.5 ADJUSTING PROGRAM INFORMATION DURING PRODUCTION RUN

During production or program run, you might need to adjust position information without stopping program execution.

Program adjust allows you to adjust positional offsets. A positional offset is a value that specifies how much of a difference there is between the current positional value and the positional value you want. It is specified for the x, y, z, w, p, and r position components.

Program adjust also allows you to adjust the linear speed and the joint speed of the program.

Program Adjust Schedules

When you make program adjustments, the changes you make are grouped together into a program adjust schedule. You can use as many as ten program adjust schedules to adjust program information during program or production run.

The program adjust schedule contains

- A **number** you assign to **identify the schedule**
- The **name of the program** being adjusted
- The **starting and ending line numbers** to be affected by the adjustment
- The **positional offset value** in x, y, z, w, p, and r
- A **linear robot speed**
- A **joint robot speed**

Program Adjust Guidelines

Use the following program adjust sequence as a guideline for your program adjustments:

1. Edit the schedule to make the program adjustments you want. Use Procedure 7-16 .
2. ENABLE the program adjust schedule. The changes will take effect as soon as the robot motion system can process the new information. Allow for one complete cycle through the program after enabling the adjustment to ensure that all positions are adjusted.
3. **To remove the adjustment**, DISABLE the offset. Allow for one complete cycle of the program for all positions to disable the adjustments.
4. **To make the adjustment a permanent adjustment**, clear the schedule (CLEAR_ADJ) and confirm. This will lock in the adjustment to the program and reset the program adjustment values to zero.
5. **To make additional adjustments**, clear the schedule (CLEAR_ADJ) and confirm. This will lock in the adjustments to the program and reset the program adjustment values to zero.
6. **If the offset is a temporary adjustment**, continue to use the offset until the temporary condition no longer exists and then DISABLE the offset.
7. **If an additional offset is required in the program**, clear the schedule (CLEAR_ADJ) and confirm. Repeat the adjustment procedure, starting with Step 1.

Use Procedure 7-16 to adjust program information during program or production run.

Procedure 7-16 Adjusting Programs During Program or Production Run

- Condition** ■ The program you want to adjust is currently selected.
- Step**
- 1 Press MENUS.
 - 2 Select UTILITIES.
 - 3 Press F1, [TYPE].
 - 4 Select Prog Adjust. You will see a screen similar to the following.

UTILITIES Prog Adj			WORLD	100%
Program	Lines	Status	1/10	
1 PRG123	22-29	EDIT		
2 PRG123	39-49	ENABLED		
3 PRG34	10-14	DISABLED		
4 PRG45567	123-456	DISABLED		
5 *****	0-0	*****		
6 *****	0-0	*****		
7 *****	0-0	*****		
8 *****	0-0	*****		
9 *****	0-0	*****		
10 *****	0-0	*****		
[TYPE] DETAIL			>	
COPY CLR_ADJ CLR_ALL			>	

- 5 Select a program and line numbers to adjust.

To adjust program parameters for the current program if it is not listed on the screen, select an unused schedule (*****), and press DETAIL. The current program name will be entered automatically.

- 6 Press F2, DETAIL. You will see a screen similar to the following.

UTILITIES Prog Adj			WORLD	100%
			1/11	
Current Schedule:	1	Status:	EDIT	
1 Program name:	PRG123			
2 Starting line number:	22			
3 Ending line number:	29			
4 X adjustment:	5.000 mm			
5 Y adjustment:	0.000 mm			
6 Z adjustment:	-2.500 mm			
7 W adjustment:	0.000 dg			
8 P adjustment:	0.000 dg			
9 R adjustment:	0.000 dg			
10 Linear speed:	0 mm/sec			
11 Joint speed:	0 %			
[TYPE] UNITS SCHED ENABLE			>	
COPY CLR_ADJ CLR_ALL			>	

7 To display the DETAIL for a different program, press ENTER.

- a** Move the cursor to select a method of naming the program: Upper Case, Lower Case, Punctuation, or Options.
- b** Press the function keys whose labels correspond to the name of the program you want. These labels vary depending on the naming method you chose in Step a. **To delete a character, press BACK SPACE.**

For example, if you chose Upper Case or Lower Case, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the program name field. Press the right arrow key to move the cursor to the next space. Continue until the entire program name is displayed.

- c** When you are finished, press ENTER.

Enter Schedule Number:

8 To display the DETAIL for a different schedule, press F3, SCHED.**9 To display the next schedule DETAIL automatically, press SHIFT and F3, SCHED.**

2 Starting line number

- 10** Select item 2 and type the starting line number where you want the changes to take effect. If you enter a line number that is out of the range of program lines, the last line number in the program will be entered automatically.

3 Ending line number

- 11** Select item 3 and type the ending line number where you want the changes to stop taking effect. If you enter a line number that is out of the range of program lines, the last line number in the program will be entered automatically.

NOTE The ending line must be greater than or equal to the starting line number specified in item 2. **To change only one line number, the ending line number must be the same as the starting line number.**

- 12 To select the units (inches or millimeters) for x, y, and z offsets, press F2, UNITS.**

4 X adjustment

5 Y adjustment

6 Z adjustment

- 13 To adjust x, y, and z offsets, select the item and type the new offset value. To indicate negative offsets, use the minus sign. The range of x, y, and z offsets is +/- 26.00 mm.**

7 W adjustment

8 P adjustment

9 R adjustment

- 14 To adjust w, p, and r offsets, select the item and type the new offset value. These offsets are always shown in degrees. To indicate negative offsets, use the minus sign. The range of w, p and r offsets is +/- .500 dg.**

10 Linear speed

- 15 To change linear speed, select Linear Speed and type the new speed value. A value of 0 indicates no change.**

11 Joint speed

- 16 To change joint speed, select Joint Speed and type the new speed value. A value of 0 indicates no change.**

```
Clear all XYZWPR/speed ad-
justments? [NO]
                YES  NO
```

```
Schedule cleared successfully
```

- 17 If you want to clear the x, y, z, w, p, and r portion of this schedule,** press NEXT, >, and then press F2, CLR_ADJ. This
- Changes the x, y, z, w, p, and r offset values to 0
 - Retains the program name and line numbers
 - Does not change the positional information in the program
 - **To clear,** press F4, YES.
 - **To cancel,** press F5, NO.

- 18 When you are finished adjusting program parameters,** press F4, ENABLE.

This activates the program adjustments you made and changes the program. The adjustments take effect as soon as the robot motion system can process them.

NOTE If a motion instruction contains a PR[n] (position register) or INC (Incremental motion option), it will not be adjusted.

- 19 To test the adjustments if the program is not running,** refer to Section 7.2.
- 20 If you are not satisfied with the adjustments,** press F5, DISABLE. This returns the program positions to the values they had before you enabled the schedule. The disable feature takes effect as soon as the robot motion system can process it.
- If you are not satisfied with speed changes,** you must EDIT the schedule to enter new speed values and ENABLE them.

 **CAUTION**

The next step describes how to change position values in the program permanently. Be sure you want to change program values permanently before you perform this step, otherwise, unexpected results could occur.

```
Clear entire schedule? [NO]
                YES  NO
```

- 21 To save the changes to your program permanently,** press F4, ENABLE to enable the changes, then press NEXT, >, and then press F3, CLR_ALL. This
- Activates the program adjustments you made and makes permanent changes to the program
 - Changes the x, y, z, w, p, and r offset values displayed on the screen to 0
 - Changes the linear speed and joint speed values displayed on the screen to 0
 - Clears the program name and line numbers displayed on the screen
 - **To clear,** press F4, YES.
 - **To cancel,** press F5, NO.

```
Enter schedule number to copy to:
```

- 22 To copy schedule information from one program to another,** press NEXT, >, and then press F1, COPY. Type the schedule number you want to copy to a specified schedule and press ENTER.

```
Schedule is initialized Continue?
```

- 23 To continue with the copy,** press F4, YES. The program will be copied.

7.6 MAINTENANCE AND REPAIR

You can use macro commands that appear on the MANUAL FCTNS menu to perform maintenance and repair procedures during production. Refer to Chapter 3 for more information about how to set up macro commands.

Use Procedure 7-17 to perform a manual function.

Procedure 7-17 Using the MANUAL FCTNS Menu

Condition ■ Macro commands that perform maintenance and repair functions have been set up to appear on the Refer to Chapter 8.

- Step**
- 1 Press MENUS.
 - 2 Select MANUAL FCTNS.
 - 3 Press F1, [TYPE].
 - 4 Select Macros. See the following screen for an example.

Manual Macros		JOINT	10 %
			1/7
Instruction			
1	OPEN HAND 1	MF[1]
2	CLOSE HAND 1	MF[1]
3	RELAX HAND 1	MF[1]
4	OPEN HAND 2	MF[2]
5	CLOSE HAND 2	MF[2]
6	RELAX HAND 2	MF[2]
7	GO TO REPAIR POS	MF[2]
[TYPE]		EXEC	

- 5 Move the cursor to the instruction you want to execute.
- 6 Continuously press and hold in the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
- 7 Press and hold the SHIFT key and press F3, EXEC. The F3 key can be released, but the SHIFT key must be held continuously until the instruction has finished executing.

8

STATUS DISPLAYS AND INDICATORS

Topics In This Chapter		Page
Status Indicators	Teach pendant and standard operator panel status indicators show various conditions of the system	8-2
	• Teach Pendant Status Indicators	8-2
	• Standard Operator Panel Status Indicators	8-3
User Screen Status	The user screen displays messages sent to the user from a running program	8-5
Register Status	The DATA Register screen displays the current value of each register in the system	8-6
Position Register Status	The DATA Position Reg screen displays the current value of each position register in the system	8-8
Pallet Registers	Pallet registers allow you to define the number of layers, rows, and columns that will be used to palletize	8-11
System Variable Status	The SYSTEM Variables status screen displays all system variables	8-13
Safety Signal Status	The STATUS Safety signal screen displays the status of safety-related control signals coming into the controller	8-15
Program Timer Status	The program timer screen displays the execution time for a program that contains TIMER instructions	8-18
System Timer	The system timer screen displays lengths of time for turning on system power, running time, waiting time, and welding time	8-20
Clock	The clock menu displays the current data and time	8-22
Version Identification Status	The STATUS Version ID screen displays information specific to your controller	8-23
Memory Status	The memory status screen displays memory use status and hardware memory structure	8-26
Position Status	The POSITION screen displays positional information in joint angles or Cartesian coordinates	8-28
Turn Number Display	Turn number display specifies the turn number displayed on the teach pendant screen	8-30
	• Usual Configuration	8-31
	• \$SCR_GRP[group].\$turn_axis[i] System Variable	8-33
Execution History	The program execution history function records the execution history of the most recently executed or halted program and enables checking of the execution history when the program terminates or halts	8-34
SOP I/O Status	The SOP I/O screen indicates the status of the standard operator panel signals	8-36
Duty Diagnosis	The duty diagnosis feature provides servomotor duty cycle information.	8-38
	• Duty Value	8-38
	• Status	8-39

Status displays provide information on the state of the system. You use this information to monitor, correct, or change how the system operates. Status indicators are LEDs on the teach pendant or operator panel that indicate various conditions.

8.1 STATUS INDICATORS

Teach pendant and standard operator panel status indicators show various conditions of the system. Your system can also have other indicators on user operator panels. See your supervisor for information about user operator panel indicators.

8.1.1 Teach Pendant Status Indicators

Teach pendant status indicators indicate the system condition when you are using the teach pendant to control the system.

Figure 8-1 shows the teach pendant status indicators. Table 8-1 lists and describes each teach pendant status indicator.

Figure 8-1. Teach Pendant Status Indicators

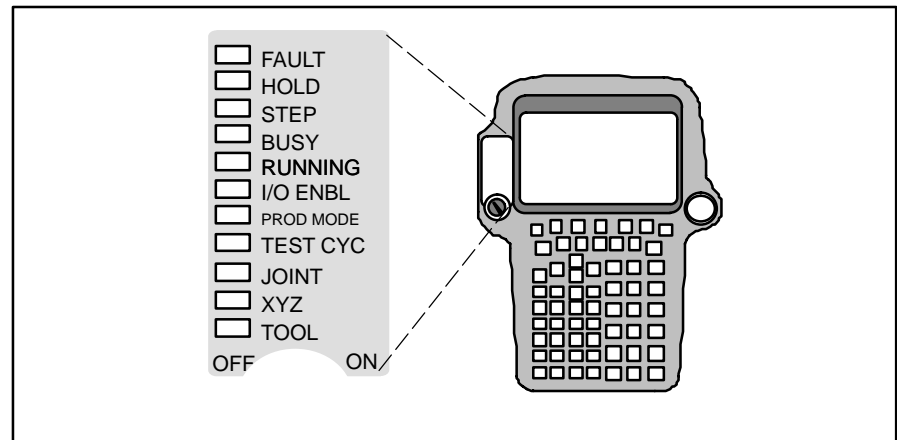


Table 8-1. Teach Pendant Status Indicators

INDICATOR	DESCRIPTION
FAULT	Indicates that a fault condition has occurred.
HOLD	Indicates that the robot is in a hold condition. HOLD is not on continuously during a hold condition.
STEP	Indicates that the robot is in step mode.
BUSY	Indicates that the controller is processing information.
RUNNING	Indicates that a program is being executed.
I/O ENBL	Indicates that I/O is enabled.
PROD MODE	Indicates that the system is in production mode and CYCLE START will start the process.
TEST CYC	Indicates that the system is in test cycle mode.
JOINT	Indicates that the current jog coordinate system is JOINT.
XYZ	Indicates that the current jog coordinate system is User or Jog frame.
TOOL	Indicates that the current jog coordinate system is TOOL. The jog speed can be changed to maximum 100 % by pressing override key.

**8.1.2
Standard Operator
Panel Status Indicators**

Standard operator panel status indicators indicate the system condition when you are using the operator panel to control the system.

Figure 8-2 and Figure 8-3 show the standard operator panel status indicators. Table 8-2 lists and describes each operator panel status indicator.

Figure 8-2. R-J2 Controller (i-Size) Standard Operator Panel

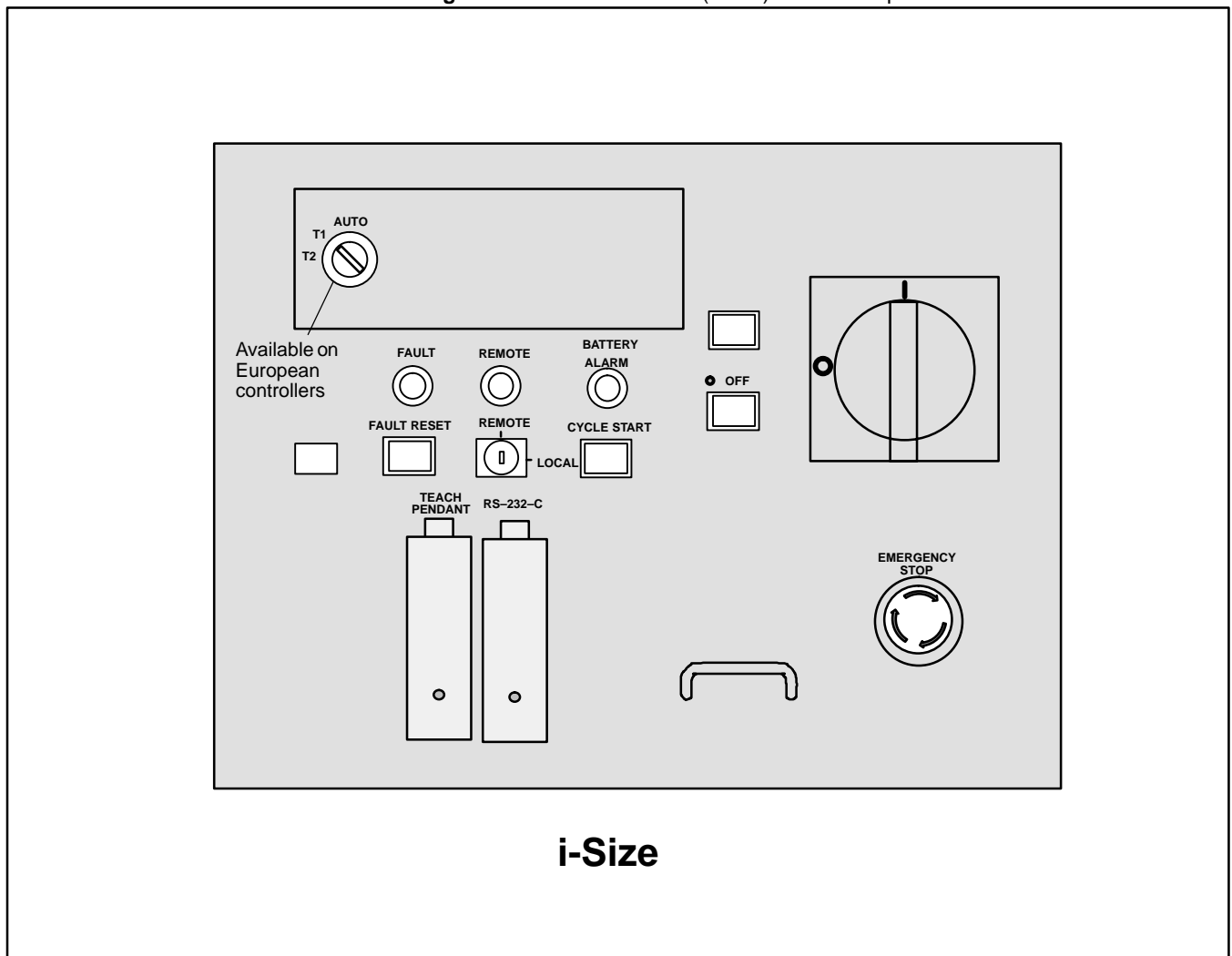


Figure 8-3. R-J2 Controller (B-Size) Standard Operator Panel

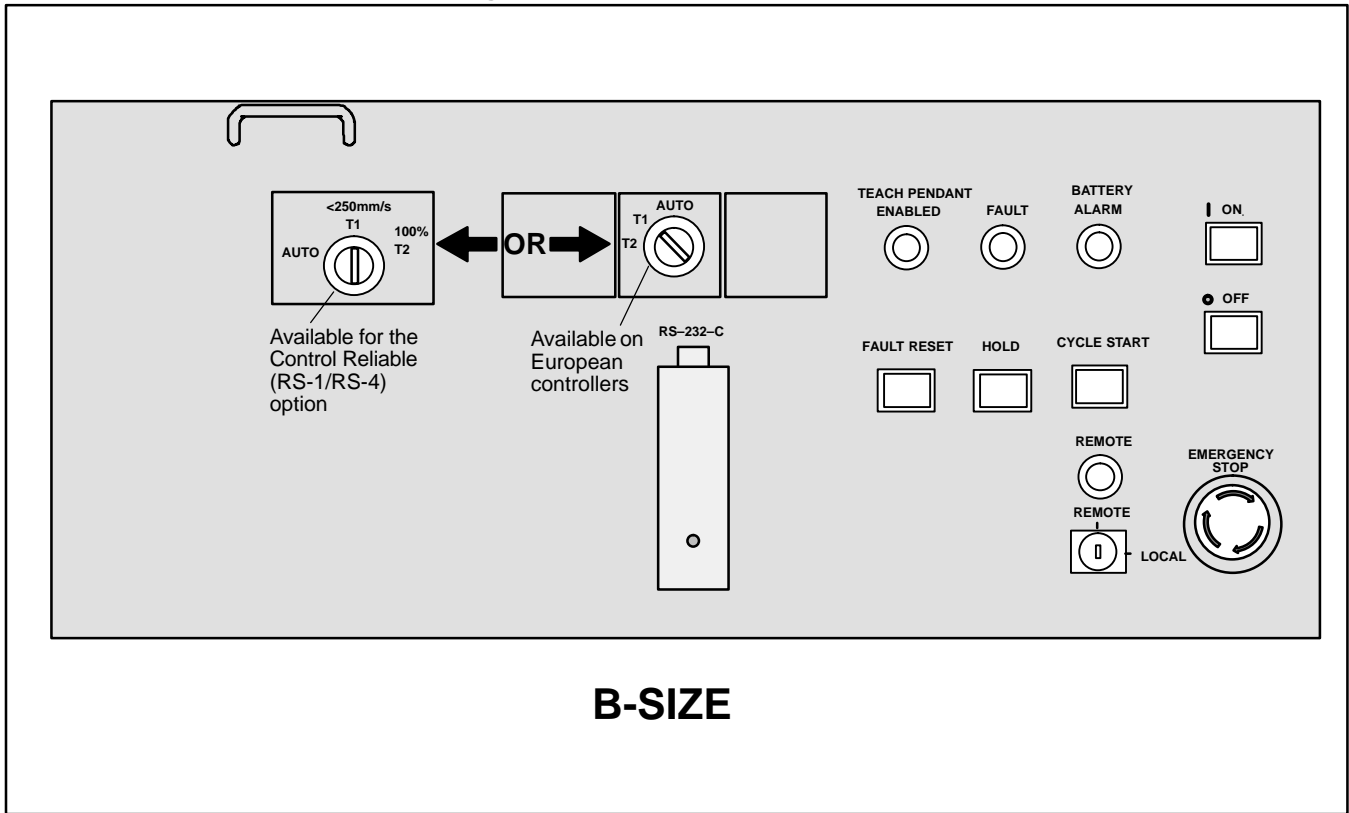


Table 8-2. Standard Operator Panel Status Indicators

INDICATOR	DESCRIPTION
BATTERY ALARM	Indicates that the voltage of the backup battery is low.
TEACH PENDANT ENABLED	Indicates that the teach pendant is enabled and has motion control.
FAULT	Indicates a fault condition has occurred.
REMOTE	Indicates that neither the teach pendant nor the operator panel have motion control. Some remote device has motion control.

8.2 USER SCREEN STATUS

The user screen displays messages sent to the user from a running program. You cannot use this screen to change information.

User messages are controlled using the MESSAGE instruction in your program. Each time the MESSAGE instruction is used, one line containing up to and including 23 characters, is written to the user screen.

A maximum of nine message lines can be displayed. If more than nine message lines are used, the tenth line is added to the bottom of the screen and the top line scrolls off. Refer to Chapter 6 for information on the message instruction.

Use Procedure 8-1 to display the user screen.

Procedure 8-1 Displaying the User Screen

- Step**
- 1 Press MENUS.
 - 2 Select USER. You will see a screen similar to the following.

NOTE This screen is blank if no messages were written. The screen saves messages even after the program has aborted.

```

TPIF-014 Teach pendant is disabled
TEST                LINE 1                RUNNING
USER                WORLD 10 %

THE_SYSTEM_HAS_POWERED
UP_SUCCESSFULLY

MOVE_THE_ROBOT_TO_HOME
POSITION_BEFORE_RUNNING
PRODUCTION

THE_ROBOT_IS_AT_THE
HOME_POSITION

```

8.3 REGISTER STATUS

The DATA Register screen displays the current value of each register in the system. You can change the value of any register and add comments using the register screen. Refer to Section 6.7 for information on registers. Use Procedure 8-2 to display the register screen.

Procedure 8-2 Displaying and Setting Registers

- Step**
- 1 Press DATA.
 - 2 Press F1, [TYPE].
 - 3 Select Registers. You will see a screen similar to the following.

Data Registers		JOINT	10 %
			1/32
R[1:]	=0	
R[2:]	=0	
R[3:]	=0	
R[4:]	=0	
R[5:]	=0	
R[6:]	=0	
R[7:]	=0	
R[8:]	=0	
R[9:]	=0	
R[10:]	=0	
[TYPE]			

CAUTION

Registers are used in programs. Do not modify register values unless you are sure how the register is used in the system, otherwise you could affect how programs are executed.

- 4 To add a comment
 - a Move the cursor to the register number and press ENTER.
 - b Move the cursor to select a method of naming the comment: Upper Case, Lower Case, Punctuation Options.
 - c Press the function keys whose labels correspond to the name you want to give to the comment. These labels vary depending on the naming method you chose in Step b.

For example, if you chose Upper Case, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire comment is displayed. **To delete a character**, press BACK SPACE.
 - d When you are finished, press ENTER.

R[1: XXXXXXXXXX]

R[1:Counter] = 0

- 5** To change the value of the register
 - a** Move the cursor to the register value.
 - b** Type the new value and press ENTER.
- 6** To save the register values to a file
 - a** Press FCTN.
 - b** Select SAVE. The registers will be saved to the file, NUMREG.VR, on the default device. Refer to Chapter 9 for information on setting the device.

8.4 POSITION REGISTER STATUS

The DATA Position Reg screen displays the current value of each position register in the system. You can change the value of any position register and add comments using the DATA Position Reg screen. For information on position registers, refer to Section 6.8.

If your system is configured to have more than one group, you can set the group mask when you use any position register instruction. The group mask allows you to use function keys to specify:

- Whether the group mask will be used. If the group mask is not used, the position register instruction affects the default group only.
- The group or groups that the position register instruction will affect.

Use Procedure 8-3 to display the position register screen.

Procedure 8-3 Displaying and Setting Position Registers

- Step**
- 1 Press DATA.
 - 2 Press F1, [TYPE].
 - 3 Select Position Reg. You will see a screen similar to the following.

DATA Position Reg		JOINT	10 %
			1/16
PR[1:	[REDACTED]] =R	
PR[2:] =*	
PR[3:] =*	
PR[4:] =*	
PR[5:] =*	
PR[6:] =*	
PR[7:] =*	
PR[8:] =*	
PR[9:] =*	
PR[10:] =*	
[TYPE]		RECORD	POSITION CLEAR

R indicates the position has been recorded.

* indicates the position has not been recorded.

CAUTION

Position registers are used in programs. Do not modify position register values unless you are sure how the position register is used in the system; otherwise, you could affect how programs are executed.

4 To add a comment

PR[1: ■]

- a** Move the cursor to the position register number and press ENTER.
- b** Move the cursor to select a method of naming the comment: Words, Upper Case, Lower Case, or Options.
- c** Press the function keys whose labels correspond to the name you want to give to the comment. These labels vary depending on the naming method you chose in Step b.

For example, if you chose Upper Case, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire comment is displayed.

To delete a character, press BACK SPACE.

- d** When you are finished, press ENTER.

5 To change the value of the position register

PR[1: Home] = R

- a** Move the cursor to the position register value.
- b** Enter the new value by **recording a position** (Step 6), or **entering positional information** (Step 7).

6 To record a position,

- a** Press and hold the DEADMAN switch and turn on the teach pendant.
- b** Jog the robot to the position you want.
- c** Hold down the SHIFT key and press F3, RECORD. The * (asterisk) will change to an R to indicate the position has been recorded.

The user frame, UF, and the tool frame, UT, will be set to 15 (FHex), which indicates that the currently active user frame and tool frame will be used. Refer to Section 3.8 for information on setting up frames.

NOTE Recording position registers in a multiple motion group system records position values for **ALL** axes, regardless of the default group mask.

7 To enter positional information manually,

- a** Press F4, POSITION. You will see a screen similar to the following.

Position Detail					
PR[1]	UF:F	UT:F		CONF:N	0 0
X	0.500	mm	W	0.00	deg
Y	1.320	mm	P	90.00	deg
Z	0.750	mm	R	0.00	deg
DATA Position Reg					
PR[1]	=R	1/10
PR[2:]	=*	
PR[3:]	=*	
PR[4:]	=*	
PR[5:]	=*	
PR[6:]	=*	
Enter value					
GROUP	PAGE	CONFIG	DONE	[REPRE]	

R indicates the position has been recorded.

* indicates the position has not been recorded.

- b To change the format of the position from Cartesian coordinates to joint angles or from joint angles to Cartesian coordinates,** press F5, [REPRE] and select the coordinate system. The proper joint angles or Cartesian coordinates will be displayed. The position is automatically converted.

NOTE Joint angles are useful for zero-positioning the robot or for controlling the motion of a positioning table.

- c To change a position component,** move the cursor to the component, type the value, and press ENTER.
- d To change the motion group number,** press F1, GROUP, type the group number, and press ENTER. This only applies to systems that have been set up for multiple groups.
- e To change the configuration,** press F3, CONFIG. Select the proper configuration by pressing the up or down arrow key.
- f To display the extended axis position information,** press F2, PAGE. This only applies to systems that include extended axes.
- g** When you are finished, press F4, DONE.
- 8** To clear a position register press F5, CLEAR. This converts all positional information to all asterisks (*****).
- 9** To save the position register values to a file
- a** Press FCTN.
- b** Select SAVE. The position registers will be saved to the file, POSREG.VR, on the default device. Refer to Chapter 8 for information on setting the device.

8.5 PALLET REGISTERS

Pallet registers allow you to define the number of layers, rows, and columns that will be used to palletize. There are 32 pallet registers available.

Procedure 8-4 Displaying the Pallet Register Screen

- Step**
- 1 Press MENUS.
 - 2 Select DATA.
 - 3 Press F1, [TYPE].
 - 4 Select Pallet Reg. You will see a screen similar to the following.

DATA	Pallet Reg	JOINT	10 %
PL[1		= [1, 1, 1]	
PL[2:] = [1, 1, 1]	
PL[3:] = [1, 1, 1]	
PL[4:] = [1, 1, 1]	
PL[5:] = [1, 1, 1]	
PL[6:] = [1, 1, 1]	
PL[7:] = [1, 1, 1]	
PL[8:] = [1, 1, 1]	
PL[9:] = [1, 1, 1]	
PL[10:] = [1, 1, 1]	
[TYPE]		POSITION	

CAUTION

Pallet registers are used in programs. Do not modify pallet register values unless you are sure how the pallet register is used in the system; otherwise, you could affect how programs are executed.

5 To add a comment

PL[1: ██████████]

- a** Move the cursor to the pallet register number and press ENTER.
- b** Move the cursor to select a method of naming the comment.
- c** Press the function keys whose labels correspond to the name you want to give to the comment. These labels vary depending on the naming method you chose in Step b.

For example, if you chose Alphabet, press a function key corresponding to the first letter. Press that key until the letter you want is displayed in the comment field. Press the right arrow key to move the cursor to the next space. Continue until the entire comment is displayed.

6 To delete a character, press BACK SPACE and then press ENTER.**7** To change the value of the pallet register

PL[1: Home] = [1, ██████████]

- a** Move the cursor to the pallet register value.
- b** Enter the new value.

8.6 SYSTEM VARIABLE STATUS

The SYSTEM Variables status screen displays all system variables. You can change the value of several system variables using this screen. You can also change the value of a system variable in a program using the Parameter name instruction. Refer to Section 6.12.7. Use Procedure 8-5 to display and set system variables.

Procedure 8-5 Displaying and Setting System Variables



WARNING


System variables control how the robot and controller operate. Do not set system variables unless you are certain of their effect; otherwise, you could disrupt the normal operation of the robot and controller.

- Step**
- 1 Press MENUS.
 - 2 Select SYSTEM.
 - 3 Press F1, [TYPE].
 - 4 Select Variables. You will see a screen similar to the following.

SYSTEM Variables		JOINT 50%
		1/129
1	\$ANGTOL	[9] of REAL
2	\$APPLICATION	[3] of STRING [21]
3	\$AP_MAXAX	0
4	\$AP_PLUGGED	2
5	\$AP_TOTALAX	16777216
6	\$AP_USENUM	[32] of BYTE
7	\$ASCII_SAVE	FALSE
8	\$AUTOINIT	2
9	\$BLT	0
10	\$CHECKCONFIG	FALSE
[TYPE]		

- 5 **To change the value of a system variable**
 - a Move the cursor to the variable you want to change.
 - b Type the new value.
 - c Press ENTER, or press a function key as prompted.
- 6 **If the variable is an array**, a list of array elements is displayed or if the variable is a structure, a list of fields is displayed.
 - a Move the cursor to the element or field you want to set and press ENTER.
 - b Press PREV to return to the top level SYSTEM Variables screen.
 - c Enter the necessary information.

- 7 To save the variables to a file
 - a From any of the SYSTEM Variables screens, press FCTN.
 - b Select SAVE. All the system variables will be saved to the file, SYSVARS.SV, on the default device. Refer to Chapter 8 for information on setting the device.

 **WARNING**
You must turn off the controller and turn on the controller to use the new information; otherwise, injury or damage to equipment could occur.

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

8.7 SAFETY SIGNAL STATUS

The STATUS Safety signal screen displays the status of safety-related control signals coming into the controller.

The safety signal screen displays the current state (TRUE or FALSE) of each safety signal. You cannot change the condition of the safety signal using this screen. Table 8-3 lists and describes each safety signal. Use Procedure 8-6 to display safety signal status.

Table 8-3. Safety Signals

SAFETY SIGNAL	DESCRIPTION
SOP E-Stop	Indicates whether the EMERGENCY STOP button on the operator panel has been pressed. The status is TRUE if the operator panel EMERGENCY STOP button has been pressed.
TP E-Stop	Indicates whether the EMERGENCY STOP button on the teach pendant has been pressed. The status is TRUE if the teach pendant EMERGENCY STOP button has been pressed.
Ext E-Stop (if you have the Control Reliable (RS-1/RS-4) option, refer to the Ext E-Stop description for Control Reliable (RS-1/RS-4) option only)	Indicates whether an external emergency exists. The status is TRUE if the external emergency stop contacts are open on the emergency control PCB (EMG) and the following conditions exist: <ul style="list-style-type: none"> ● SOP E-STOP is FALSE ● TP E-Stop is FALSE ● Hand Broken is FALSE ● Overtravel is FALSE If any one of these conditions is TRUE, Ext E-Stop is displayed as FALSE even though the external emergency stop switch could be open.
Ext E-Stop (for Control Reliable (RS-1/RS-4) option only)	Indicates whether an external emergency exists. The status is TRUE if the external emergency stop contacts are open on the cell connector EES1, EES11, EES2, or EES21.
Fence Open (if you have the Control Reliable (RS-1/RS-4) option, refer to the Fence Open description for Control Reliable option only)	Indicates whether the safety fence switch is open. The status is TRUE if the safety fence terminals are open on the operator's panel PCB. This does not require the teach pendant to be enabled.
Fence Open (AUTO STOP) (for Control Reliable (RS-1/RS-4) option only)	Indicates whether the safety fence switch is open. The status is TRUE if the safety fence contacts are open on the cell connector EAS1, EAS11, EAS2, or EAS21.
TP Deadman	Indicates when either the left or right teach pendant DEADMAN switch is pressed. The status is TRUE if either DEADMAN switch is pressed.
TP Enable	Indicates whether the teach pendant ON/OFF switch is ON. The status is TRUE when the teach pendant ON/OFF switch is ON.
Hand Broken	Indicates whether the safety joint switch in the robot hand has been tripped and the hand might be damaged. The status is TRUE when the safety joint switch has been tripped.

Table 8-3. (Cont'd) Safety Signals

SAFETY SIGNAL	DESCRIPTION
Overtravel	Indicates whether the robot has moved beyond its overtravel limits. The status is TRUE when the robot has moved beyond its overtravel limits tripping the overtravel switch.
Low Air Alarm	Indicates whether the air pressure has decreased below the acceptable limit. Low Air Alarm is usually connected to an air pressure sensing device. The status is TRUE when the air pressure is below the acceptable limit. You must set the \$PARAM_GROUP[1].\$PPABN_ENBL system variable to TRUE to use this signal.
Belt Broken	Indicates whether a robot belt is broken. The status is TRUE when a robot belt is broken. This turns RDI7 on or off depending on how your system is set up. You must set the \$PARAM_GROUP[1].\$BELT_ENABLE system variable to TRUE to use this signal.
SVON Input (if you have the Control Reliable (RS-1/RS-4) option, refer to the SVON Input description for Control Reliable option only)	Indicates whether the SVON input switch is open. The status is TRUE if the SVON input terminals are open on the operation box PCB.
SVON Input (General Stop) (for Control Reliable (RS-1/RS-4) option only)	Indicates whether the SVON input switch is open. The status is TRUE if the SVON input contacts are open on the cell connector EGS1, EGS11, EGS2, or EGS21.
Servo Disconnect (for Control Reliable (RS-1/RS-4) option only)	Indicates whether the SERVO DISCONNECT input switch is open. The status is TRUE if the SERVO DISCONNECT input contacts are open on the operation box PCB TBOP4 – SD4, SD41, SD5, or SD51.
Non Teacher Enabling Device (NTED) (for Control Reliable (RS-1/RS-4) option only)	Indicates whether the NTED input switch is open. The status is TRUE if the NTED input contacts are open on CRM27 on the operation box PCB.

Procedure 8-6 Displaying Safety Signal Status

- Step**
- 1 Select MENUS.
 - 2 Select STATUS.
 - 3 Press F1, [TYPE].
 - 4 Select Safety Signal. You will see a screen similar to the following.

STATUS Safety		JOINT	10 %
	SIGNAL NAME	STATUS	1/13
1	SOP E-Stop:	TRUE	
2	TP E-Stop:	FALSE	
3	Ext E-Stop:	FALSE	
4	Fence Open:	FALSE	
5	TP Deadman:	FALSE	
6	TP Enable:	FALSE	
7	Hand Broken:	FALSE	
8	Overtravel:	FALSE	
9	Low Air Alarm:	FALSE	
10	Belt Broken:	FALSE	
11	SVON Input:	FALSE	
12	Servo Disconnect:	FALSE	
13	Non Teach Enb. Dev.	FALSE	

[TYPE]

8.8 PROGRAM TIMER STATUS

The program timer screen displays the execution time for a program that contains TIMER instructions. TIMER instructions allow you to specify in your program when you want the timer to start, stop, or reset.

Table 8-4 lists and describes each item on the program timer listing screen.

Use Procedure 8-7 to display the program timer screen.

Table 8-4. Program Timer Listing Screen

ITEM	DESCRIPTION
Timer[]	Indicates the number of the timer. You use this number in the TIMER instruction in your program. There are ten timers available.
Count	Indicates the length of time, in seconds, that the program or section of program took to execute.
Comment	Allows you to name or enter a comment about a timer.

Procedure 8-7 Displaying the Program Timer Screen

- Step**
- 1 Press MENUS.
 - 2 Select STATUS.
 - 3 Press F1, [TYPE].
 - 4 Select Prg Timer. See the following screen for an example.

```

PRG TIMER LISTING                                JOINT    10 %
                                                1/10
1 Timer[ 1]          4732 ms STYLE1
2 Timer[ 2]           0 ms *****
3 Timer[ 3]           0 ms *****
4 Timer[ 4]           0 ms *****
5 Timer[ 5]           0 ms *****
6 Timer[ 6]           0 ms *****
7 Timer[ 7]           0 ms *****
8 Timer[ 8]           0 ms *****
9 Timer[ 9]           0 ms *****
10 Timer[10]         0 ms *****

[ TYPE ]

```

5 To add or change a timer value:

- a** Move the cursor to the timer line and press ENTER.
- b** Enter a value.
- c** Press the appropriate function keys to add the value.
- d** When you are finished, press ENTER.

6 To add or change a comment:

- a** Move the cursor 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.

8.9 SYSTEM TIMER

The system timer screen displays lengths of time for turning on system power, running time, waiting time, and welding time. Table 8-5 lists and describes each item on the system timer screen.

Table 8-5. System Timer Menu Listing

ITEM	DESCRIPTION
Group	Allows you to display system timers for other groups. This item only functions if your system uses multiple groups.
Timer Type	Shows the different types of time that are counted.
Total(h)	Shows the total amount of time, in hours, for each of the timer types. These totals cannot be changed or reset.
Lap(m)	Shows the amount of time, in minutes, a single cycle for each of the timer types takes to complete. Lap counts can be turned ON or OFF, and RESET.

Procedure 8-8 Displaying the System Timer Menu

- Step**
- 1 Press MENUS.
 - 2 Select STATUS.
 - 3 Press F1, [TYPE].
 - 4 Select Sys Timer. See the following screen for an example.

STATUS		JOINT	10 %
-- System Timer --			
GROUP :	1		
Timer type	Total(h)	Lap(m)	
On Power time:	12.3	0.0 [OFF]	
Running time :	4.5	0.0 [OFF]	
Waiting time :	2.3	0.0 [OFF]	
Welding time :	1.2	0.0 [OFF]	
[TYPE]	GROUP#	RESET	ON OFF

NOTE You will not be able to change the group number unless your system uses multiple groups.

- 5 To change the system timer display to a different group, press F2, GROUP#, and enter the new group number.

6 Move the cursor to the timer type you want to turn ON or OFF.

- **To turn the lap counter ON**, press F4, ON.
- **To turn the lap counter OFF**, press F5, OFF.

7 To reset a lap counter:

- a** Move the cursor to the timer type you want to reset.
- b** Press F5, OFF to turn the timer type OFF.
- c** Press F3, RESET.
- d** Press F4, YES, to reset the lap counter.



```
Reset are you sure[NO]?  
      YES      NO  
      F4      F5
```

8.10 CLOCK

The clock menu displays the current data and time. Table 8-5 lists and describes each item on the clock screen.

Table 8-6. System Timer Menu Listing

ITEM	DESCRIPTION
Date	Displays the current date by year, month and then day.
Time	Displays the current time using a 24 hour clock. The time is displayed by hour, minute, and then seconds.

Procedure 8-9 Displaying the Clock Menu

- Step**
- 1 Press MENUS.
 - 2 Select SYSTEM.
 - 3 Press F1, [TYPE].
 - 4 Select Clock. See the following screen for an example.

SYSTEM Clock	JOINT
Clock Display	
DATE	9x/01/01
TIME	18:56:28
[TYPE]	ADJUST

- 5 To change the date or time display, press F4, ADJUST, and enter the new information.

8.11 VERSION IDENTIFICATION STATUS

The STATUS Version ID screen displays information specific to your controller. Use this information when you contact service if a problem occurs with your controller. You cannot change the information displayed on this screen. Table 8-7 lists and describes the version identification status information.

Table 8-7. Version Identification Status Items

ITEM	DESCRIPTION
SOFTWARE	Lists the software item loaded.
ID	Lists the version number of the software item loaded.

Use Procedure 8-10 to display version identification status.

Procedure 8-10 Displaying the Version Identification Status

- Step**
- 1 Press STATUS.
 - 2 Press F1, [TYPE].
 - 3 Select Version ID. You will see a screen similar to the following.

STATUS	Version ID	JOINT	10 %
			1/60
	SOFTWARE:	ID:	
1	HandlingTool	7D60	
2	S/W Order No.	9020000	
3	Controller F No.	F00000	
4	A-520-FM		
5	Servo Code	JC14.01	
6	Cart. Mot. Parameter	V1.04	
7	Joint Mot. Parameter	V1.04	
8	Boot MONITOR	V4.22	
9	Teach Pendant	7D01/09I	
10	Software Edition No.	V4.40-1	
11	R-J2 Kernel	V4.40-1	
12	R-J2 Operating System	V4.40-1	
13	Test Run	V4.40-1	
14	Override Select	V4.40-1	
15	CTRL Start Menus	V4.40-1	
16	Option Installation	V4.40-1	
17	I/O Interconnect	V4.40-1	
18	Execution History	V4.40-1	
19	V-400i/MAIN Interface	V4.40-1	
20	V-400i/MAIN Int (2)	V4.40-1	
21	User Frame	V4.40-1	
22	Power Fail Recovery	V4.40-1	
23	Mirror and Shift Fctn	V4.40-1	
24	FANUC Tool Offset	V4.40-1	
25	FANUC User Frame	V4.40-1	
26	Controller Backup	V4.40-1	
27	Core Built-ins	V4.40-1	
28	FANUC New Background	V4.40-1	
29	Basic Menus	V4.40-1	
30	HandlingTool (N. A.)	V4.40-1	
31	Analog I/O	V4.40-1	
32	Servo Code for R-J2	V4.40-1	
33	MACROS, Skip/Offset	V4.40-1	
34	Incr Instruction	V4.40-1	
35	FANUC Floppy Conn	V4.40-1	
36	FANUC Print Func	V4.40-1	
37	Multi-Tasking	V4.40-1	
38	FANUC Hour Meter	V4.40-1	
39	Position Registers	V4.40-1	
40	FANUC PNS	V4.40-1	
41	FANUC Prog Adjust	V4.40-1	
42	TCP Auto Set	V4.40-1	
43	Condition Monitor	V4.40-1	
44	M-16i (ARCMATE-120i)	V4.40-1	
45	*****	*****	
59:	*****	*****	
60:	*****	*****	
[TYPE]	SOFTWARE	MOT_ID	MOT_INF SER_PAR

NOTE The information displayed here could be different at your site.

4 Press the key that corresponds to the version ID status screen you want to display:

- **To display software version information**, press F2, SOFTWARE.
- **To display motor types for each axis**, press F3, MOT_ID. You will see a screen similar to the following.

STATUS Version IDs			JOINT	10 %
GRP:	AXIS:	MOTOR ID:		
1: 1	1	AC1-0S/3000 12A		
2: 1	2	AC1-0S/3000 12A		
3: 1	3	AC2-0S		
4: 1	4	AC3-0S		
5: 1	5	AC4-0S		
6: 1	6	AC5-0S		
7: *	*	Uninitialized		
8: *	*	Uninitialized		
9: *	*	Uninitialized		
10: *	*	Uninitialized		
[TYPE]	SOFTWARE	MOT_ID	MOT_INF	SER_PAR

- **To display the motor information for each axis**, press F4, MOT_INF. You will see a screen similar to the following.

STATUS Version IDs			JOINT	10 %
GRP:	AXIS:	MOTOR INFO:		
1: 1	1	H1 DSP1-L		
2: 1	2	H2 DSP1-M		
3: 1	3	H3 DSP2-L		
4: 1	4	H4 DSP2-M		
5: 1	5	H5 DSP3-L		
6: 1	6	H6 DSP3-M		
7: *	*	Uninitialized		
8: *	*	Uninitialized		
9: *	*	Uninitialized		
10: *	*	Uninitialized		
[TYPE]	SOFTWARE	MOT_ID	MOT_INF	SER_PAR

- **To display the servo parameters for each axis**, press F5, SER_PAR. You will see a screen similar to the following.

STATUS Version IDs			JOINT	10 %
GRP:	AXIS:	SERVO PARAM ID:		
1: 1	1	P01.02		
2: 1	2	P01.02		
3: 1	3	P01.02		
4: 1	4	P01.02		
5: 1	5	P01.02		
6: 1	6	P01.02		
7: *	*	Uninitialized		
8: *	*	Uninitialized		
9: *	*	Uninitialized		
10: *	*	Uninitialized		
[TYPE]	SOFTWARE	MOT_ID	MOT_INF	SER_PAR

8.12 MEMORY STATUS

The memory status screen displays memory use status and hardware memory structure. Table 8-8 lists and describes each memory status item. Use Procedure 8-11 to display memory status.

Table 8-8. Memory Status

MEMORY STATUS		DESCRIPTION
Pools	TPP	Indicates the amount of memory for user program: teach pendant programs and KAREL programs.
	PERM	Indicates the amount of memory for the system variables, the registers and the position registers.
	SYSTEM	Indicates the amount of memory for a part of the system software.
	IMAGE	Indicates the amount of memory for a part of system variables and a part of KAREL programs.
	TEMP	Indicates the amount of memory for the system software work area.
Hardware	F-ROM	Displays storage capacity of the F-ROM module used in control unit.
	C-MOS	Displays storage capacity of the C-MOS module used in control unit.
	D-RAM	Displays storage capacity of the D-RAM module used in control unit.

Procedure 8-11 Displaying Memory Status

- Step**
- 1 Press **MENUS**.
 - 2 Press **F1**, [**TYPE**].
 - 3 Select **Memory**. You will see a screen similar to the following.

STATUS Memory		JOINT 10%	
		Total	Available
Pools			
TPP	CMOS	1024.0 KB	1002.9 KB
PERM	CMOS	2023.8 KB	918.0 KB
TEMP	DRAM	1598.9 KB	1046.6 KB
Description:			
TPP:	Used by .TP, .MR, .JB, .PR		
PERM:	Used by .VR, RD:, Options		
TEMP:	Used by .PC, .VR, Options		
[TYPE]		DETAIL	HELP

4 Press F2, DETAIL. See the following screen for an example.

STATUS Memory		JOINT 10%	
Pools -----			
TPP	1024.0 KB	1002.9 KB	1002.9 KB
PERM	2023.8 KB	918.0 KB	918.0 KB
SYSTEM	1138.4 KB	130.0 KB	130.3 KB
IMAGE	255.9 KB	66.9 KB	66.5 KB
TEMP	1598.9 KB	1046.6 KB	1046.6 KB
Hardware -----			
FROM	2.0 MB		
DRAM	4.0 MB		
CMOS	2.0 MB		
[TYPE]	BASIC		HELP

To return to the basic screen from the detail screen, press F2, BASIC. To get the brief description for each memory area, press F5, HELP.

NOTE In these screens, you cannot change the memory assignment.

8.13 POSITION STATUS

The POSITION screen displays positional information in joint angles or Cartesian coordinates. The positional information on this screen is updated continuously when the robot moves. You cannot change the displayed information using this screen. Refer to Section 6.3.2 for a description of positional information.

NOTE E1, E2, and E3 indicate extended axis positional information if extended axes are installed in your system.

Joint

The joint screen displays positional information in degrees for each robot axis. Tool indicates the number of the active tool frame.

User

The user screen displays positional information in Cartesian coordinates based on the user frame. Tool indicates the number of the active tool frame. Frame indicates the number of the active user frame.

World

The world screen displays positional information in Cartesian coordinates based on the world frame. Tool indicates the number of the active tool frame.

Use Procedure 8-12 to display position status.

Procedure 8-12 Displaying Position Status

- Step**
- 1 Press MENUS.
 - 2 Select POSITION.
 - 3 Select the appropriate coordinate system.
 - **For joint**, press F2, JNT. You will see a screen similar to the following.

POSITION		JOINT		10 %
Joint			Tool: 1	
J1:	.001	J2:	10.028	J3: -35.025
J4:	-.000	J5:	34.998	J6: .001
E1:		E2:		E3:
[TYPE]	JNT	USER	WORLD	

NOTE E1:, E2:, and E3 are displayed only if you have extended axes.

- **For user**, press F3, USER. You will see a screen similar to the following.

POSITION		USER	10 %
User	Frame: 0	Tool: 1	
Configuration: F, 0, 0, 0			
x: 1906.256	y: .041	z: 361.121	
w: 178.752	p: -89.963	r: 1.249	
E1: .001	E2: .001	E3: .001	
[TYPE]	JNT	USER	WORLD

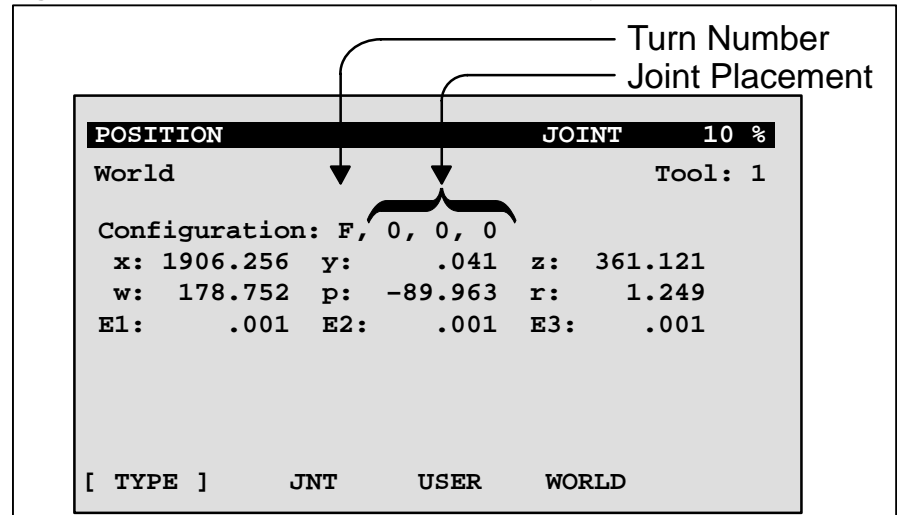
- **For world**, press F4, WORLD. You will see a screen similar to the following.

POSITION		WORLD	10 %
World		Tool: 1	
Configuration: F, 0, 0, 0			
x: 1906.256	y: .041	z: 361.121	
w: 178.752	p: -89.963	r: 1.249	
E1: .001	E2: .001	E3: .001	
[TYPE]	JNT	USER	WORLD

8.14 TURN NUMBER DISPLAY

Turn number display specifies the turn number displayed on the teach pendant screen. Figure 8-4 shows an example of where joint placement and turn number information is displayed on the STATUS Position screen. Refer to Section 8.13 for more information.

Figure 8-4. Turn Number and Joint Placement Display on Position Screen



- For most robot models, the position data is usually represented in the format shown in Section 8.14.1.
- For some robot models the system variable \$SCR_GRP[group].\$turn_axis[i] must be used to set position data. Refer to Section 8.14.2.

8.14.1 Usual Configuration

For the most robot models, the values of the system variable are as follows (with some exceptions):

```
$SCR_GRP[group].$TURN_AXIS[1]=4
$SCR_GRP[group].$TURN_AXIS[2]=5
$SCR_GRP[group].$TURN_AXIS[3]=6
```

See Figure 8-5.

Figure 8-5. Turn Number Display Configuration

(F, L, U, T, J4 J5 J6 0, 0, 0)	
Joint Placement	Turn Number
Joint Placement	
{ FLIP NOFLIP	Upward and downward placement of the wrist
{ LEFT RIGHT	Right and left placement of the arm For horizontally articulated robots only
{ UP DOWN	Upward and downward placement of the arm
{ FRONT BACK	Backward and forward placement of the arm
Turn number	
{ 1: 180° ~ 539° 0: -179° ~ 179° -1: -539° ~ -180°	

In general, when the robot arm can reach the same TCP with the arm bent differently, a configuration identifier is needed to specify the joint placement. The joint placement varies for fully articulated robots, such as the M-710i, and horizontally articulated robots, such as the A-520i. See Figure 8-6 and Figure 8-7.

Figure 8-6. Joint Placement Configuration Examples for Fully Articulated Robots

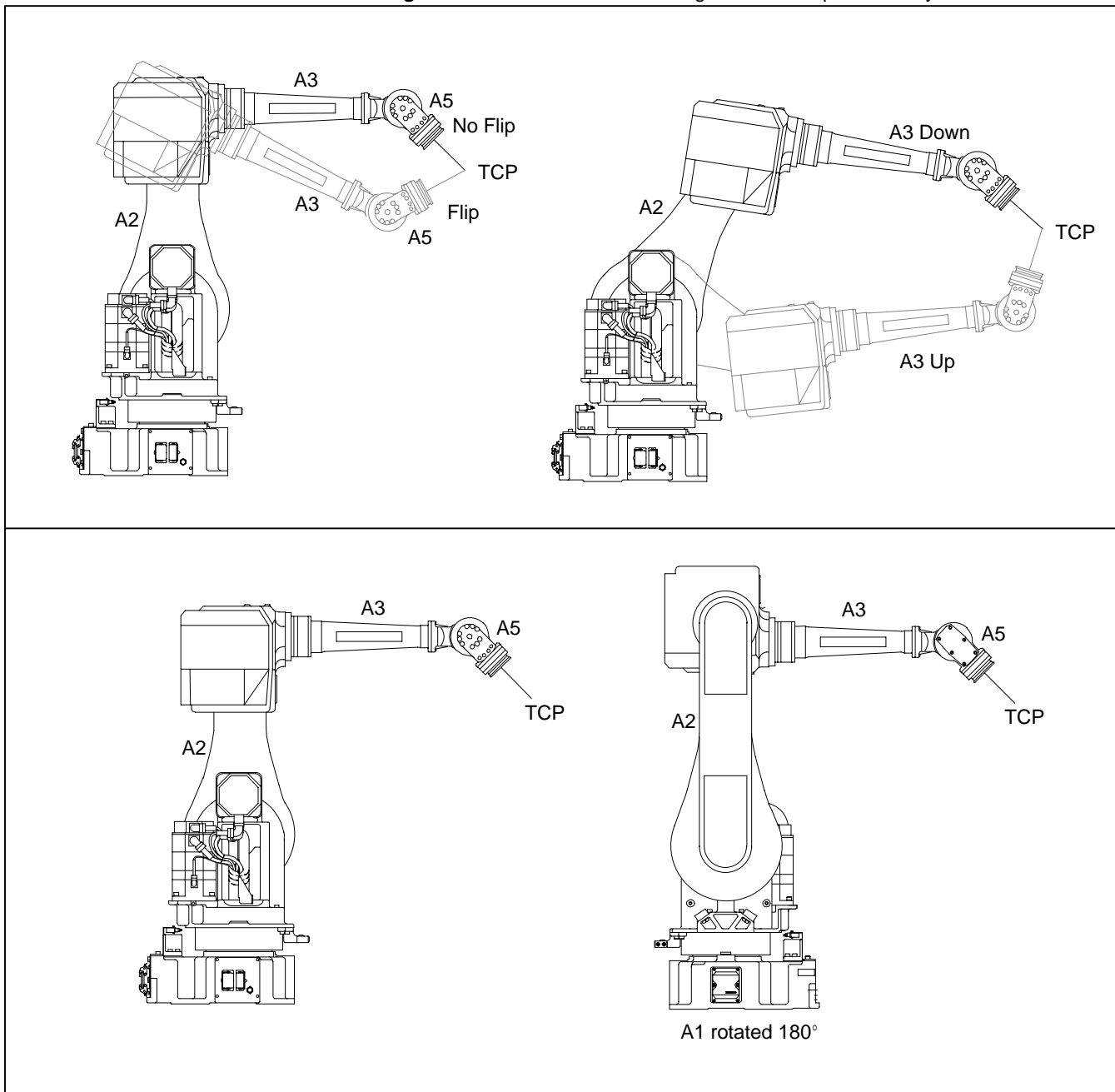
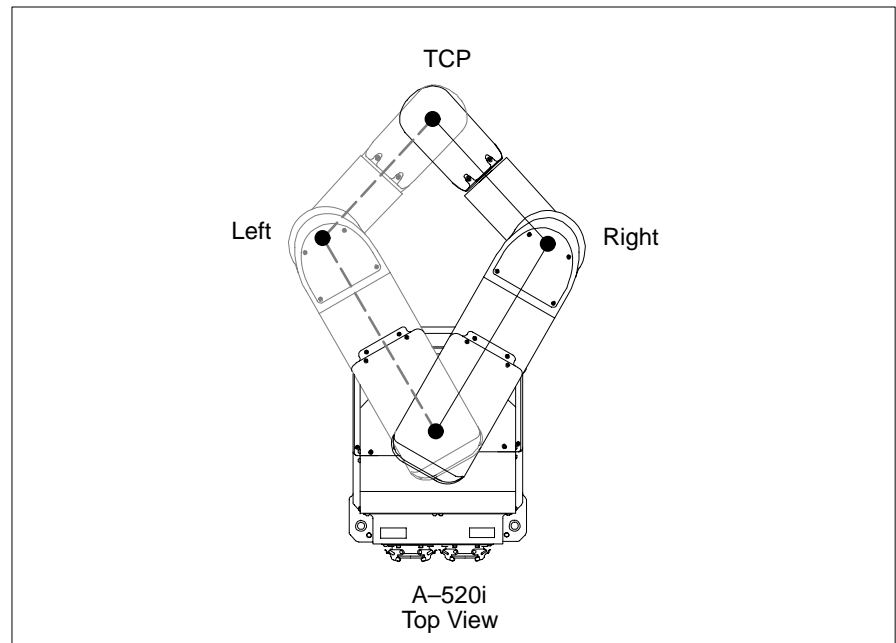


Figure 8-7. Joint Placement Configuration Examples for Horizontally Articulated Robots

8.14.2

`$SCR_GRP[group]` `.$turn_axis[i]` System Variable

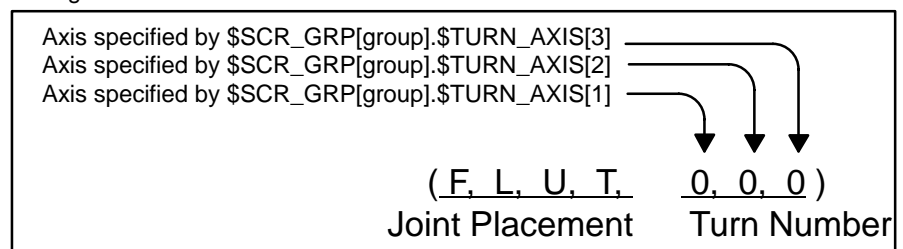
If a robot model has a turn number for the J1 axis, the usual configuration shown in Figure 8-5 will not represent the positional data. To represent the positional data in this case, the system will set the system variable `$SCR_GRP[group].$turn_axis[i]` (where $i = 1, 2, \text{ or } 3$) to the appropriate value as shown in Figure 8-8.

For models with a turn number for the J1 axis, such as the S-420iF, the values of the system variable are as follows:

```
$SCR_GRP[group].$TURN_AXIS[1]=1
```

```
$SCR_GRP[group].$TURN_AXIS[2]=4
```

```
$SCR_GRP[group].$TURN_AXIS[3]=6
```

Figure 8-8. `$SCR_GRP[group].$turn_axis[i]` for Turn Number Display Configuration

In Figure 8-8 `$SCR_GRP[group].$turn_axis[i]` specifies the turn numbers for the robot axes. (These correspond to axes J4, J5, and J6 in the usual configuration.)

8.15 EXECUTION HISTORY

The program execution history function records the execution history of the most recently executed or halted program and enables checking of the execution history when the program terminates or halts.

NOTE Execution history of a program that is currently running cannot be checked.

The information displayed for each running program is listed and described in Table 8-9.

Table 8-9. Execution History Screen

ITEM	DESCRIPTION
Program name	Displays the name of the program being executed.
Line	Displays the program line number.
Direction	Displays the direction in which the specified line number is executed: forward (FWD) or backward (BWD).
Status	Displays the execution status of the specified line number: <ul style="list-style-type: none"> • Not exec – The line was read but has not yet been executed. • Paused – While the line was being executed, the program halted (incomplete execution). • Done – The line was executed completely. • Aborted – The program was terminated.

When a program has been aborted, the information displayed in item 1 of the screen will contain a blank program name, a line number of 0, a blank direction, and a status of Aborted.

In standard execution history, up to 20 lines can be recorded. The number of lines to be recorded can be changed at controlled start. Refer to Appendix C. Information that is older than the lines recorded in the execution history is erased automatically.

When Execution History is Not Recorded

Execution history is not recorded in the following cases:

- If a macro is executed using a method other than a program (such as MANUAL FCTNS or a user key), execution history is not recorded. If a program assigned to a macro is executed from the program editing screen, the execution history is recorded with the assigned program name, not with the macro name.
- The execution history of a KAREL program is not recorded, even if a KAREL program is executed.
- The execution history of a program that is executed automatically when power is turned on is not recorded.

Use Procedure 8-13 to display the execution history screen.

Procedure 8-13 Displaying Program Execution History

- Step**
- 1 Press STATUS.
 - 2 Press F1, [TYPE].
 - 3 Select Exec-hist. You will see a screen similar to the following.

Execution history			JOINT 10 %
			1/20
Program name	Line.	Dir.	Stat.
1 █	0		Aborted
2 PROG987	9	FWD	Done
3 PROG987	8	FWD	Done
4 PROG987	7	FWD	Done
5 PROG987	6	FWD	Done
6 PROG987	5	FWD	Done
7 PROG987	4	FWD	Done
8 PROG987	3	FWD	Done
9 PROG987	2	FWD	Done
Press NEXT to display other task			
[TYPE]	NEXT	ALL_CLR	CLEAR

When a program has been aborted, the information displayed in item 1 of the screen will contain a blank program name, a line number of 0, a blank direction, and a status of Aborted.

NOTE Pressing F2, NEXT, changes display from the first to the second to the third to the fourth and then back to the first program. If you display the second program, for example, you will have to press F2, NEXT, three times to display the first program.

4 To display information for any of three other simultaneously running programs, press F2, NEXT. To return to the first display, press F2, NEXT, until the program you want is displayed.

NOTE You can clear status information only for programs that have been aborted.

5 To clear the status information for the aborted program you are viewing, press and hold SHIFT and press F5, CLEAR.

6 To clear the status information for all aborted programs, press and hold SHIFT and press F4, ALL_CLR.

8.16 SOP I/O STATUS

The SOP I/O screen indicates the status of the standard operator panel signals. SOP input signals (SI) and SOP output signals (SO) correspond to internal controller software Panel Digital Input signals (PDI) and Panel Digital Output signals (PDO). Refer to Table 8-10 and Table 8-11.

Table 8-10. Standard Operator Panel Input Signals

SI	PDI	Function	Description
0	1		Not used.
1	2	FAULT RESET	Input signal is normally turned OFF, indicating that the FAULT RESET button is not being pressed.
2	3	REMOTE	Input signal is normally turned OFF, indicating that the controller is not set to remote.
3	4	HOLD	Input signal is normally turned ON, indicating that the HOLD push button is not being pressed.
4	5		Used for mode selection switch.
5	6		Used for mode selection switch.
6	7	CYCLE START	Input signal is normally turned OFF, indicating that the CYCLE START push button is not being pressed.
7-15	8-16	NOT USED	Open for additional PDI.

Table 8-11. Standard Operator Panel Output Signals

SO	PDO	Function	Description
0	1	REMOTE LED	Output signal indicates the controller is set to remote.
1	2	CYCLE START	Output signal indicates the CYCLE START button has been pressed or a program is running.
2	3	HOLD	Output signal indicates the HOLD button has been pressed or a hold condition exists.
3	4	FAULT LED	Output signal indicates a fault has occurred.
4	5	BATTERY ALARM	Output signal indicates the voltage in the battery is low.
5	6	USR LED#1	Not used.
6	7	USR LED#2	Not used.
7	8	TEACH PENDANT ENABLED	Output signal indicates the teach pendant is enabled.
8-15	9-16	NOT USED	Open for additional PDO.

Use Procedure 8-14 to display and force SOP I/O.

Procedure 8-14 Displaying and Forcing SOP I/O

- Step**
- 1 Press MENUS.
 - 2 Select I/O.
 - 3 Press F1, [TYPE].
 - 4 Select SOP. See the following screen for an example.

I/O SOP Out			JOINT	10 %
#	STATUS			
SO[1]	OFF	[]
SO[2]	OFF	[]
SO[3]	OFF	[]
SO[4]	OFF	[]
SO[5]	OFF	[]
SO[6]	OFF	[]
SO[7]	OFF	[]
SO[8]	OFF	[]
SO[9]	OFF	[]
SO[10]	OFF	[]
[TYPE]		IN/OUT	ON	OFF

To change between the display of the input and output screens, press F3, IN/OUT.

To move quickly through the information, press and hold the SHIFT key and press the down or up arrow keys.

NOTE You can only view the status of input signals. Input signals cannot be forced.

5 To force an output signal, move the cursor to the output you want to change:

- **To turn on an output signal,** press F4, ON.
- **To turn off an output signal,** press F5, OFF.

8.17 DUTY DIAGNOSIS

The duty diagnosis feature provides servomotor *duty* cycle information. Duty is the physical load on the robot servomotor.

You look at duty cycle information:

- To check whether there is a high load on a servomotor
- To diagnose a servomotor overheating problem

NOTE Duty diagnosis is available only if the option is supported on the robot. Duty diagnosis is available only for specific robots.

8.17.1 Duty Value

The duty value of a servomotor is displayed as a percentage value. The percentage is the ratio of the root mean square current of the motor to the allowable root mean square current, at a defined temperature. See Figure 8-9 for an example of the Duty Diagnosis screen.

Figure 8-9. Duty Diagnosis Screen

STATUS/Axis		WORLD	100%
		GRP[1]	
Duty Diagnosis			
Temperature	----	25.00	(C)
J1 : DUTY	----	89.35	%
J2 : DUTY	----	199.99	%
J3 : DUTY	----	1.50	%
J4 : DUTY	----	5.99	%
J5 : DUTY	----	78.56	%
J6 : DUTY	----	55.44	%
[TYPE]	REG.DIS	DUTY	[UTIL] >

Duty Value Less than 100%

A **duty value less than 100%** means that the servomotor is working within an acceptable load for the defined temperature.

The ideal operating conditions are when the duty values displayed for all joint axes are less than 100%.

Duty Value Greater than 100%

A **duty value greater than 100%** means that the load on the servomotor is above the recommended range for the defined temperature. A value greater than 100% does not necessarily mean that the motor will overheat. It does mean that the servomotor is working hard for the conditions and has the potential to overheat.

When a servomotor has a duty value greater than 100%, you should adjust the program to reduce the load on the servomotor. For example, you could add an ACC instruction to increase the acceleration time, or a WAIT instruction to slow the execution of the program. In addition, you could touchup certain positions to minimize duty, if possible.

In Figure 8-9, the J2 servomotor might overheat if the program is not adjusted.

NOTE If the servomotor for an axis overheats, the message, “SRVO-046 SERVO OVC alarm,” will be displayed on the teach pendant screen. This message is displayed **before** damage has been done to the servomotor. When you see this message, you should take actions to reduce the load on the servomotor to eliminate the overheating condition. Refer to the *FANUC Robotics SYSTEM R-J2 Controller Electrical Connection and Maintenance Manual* for more information on recovery from an overheated servomotor.

8.17.2 Duty Status

Table 8-12 lists and describes the items displayed on the Duty Diagnosis screen.

Table 8-12. Duty Diagnosis Screen Items

ITEM	DESCRIPTION
Temperature default: 25°C	Temperature is the ambient temperature, or the air temperature of your work environment. You must specify this temperature, and the units (°C or °F), to display accurate duty percentage values.
Jn : DUTY	Displays the duty percentage value for the specified servomotor. If this value is greater than 100%, the servomotor might overheat in the future.

Use Procedure 8-15 to set the temperature and display the duty status.

Procedure 8-15 Setting Temperature and Displaying Duty Status

- | | |
|-------------|--|
| Step | <ol style="list-style-type: none"> 1 Press MENUS. 2 Select STATUS. 3 Press F1, [TYPE]. 4 Select Axis. 5 Press NEXT, >, until F2, DUTY, is displayed. |
|-------------|--|

6 Press F2, DUTY. You will see a screen similar to the following.

STATUS/Axis		WORLD	100%
		GRP[1]	
Duty Diagnosis			
Temperature	----	25.00	(C)
J1 : DUTY	----	89.35	%
J2 : DUTY	----	199.99	%
J3 : DUTY	----	1.50	%
J4 : DUTY	----	5.99	%
J5 : DUTY	----	78.56	%
J6 : DUTY	----	55.44	%
[TYPE]	REG.DIS	DUTY	[UTIL] >
		C	F >

NOTE You must specify the temperature of your work environment to obtain accurate duty values. The controller does not measure this temperature for you.

- 7 Move the cursor to the temperature value and type the temperature of your work environment.
- 8 Specify the units of the temperature value:
 - a Press NEXT, >.
 - b Press the appropriate key:
 - For Celsius, press F4, C.
 - For Fahrenheit, press F5, F.
- 9 Run the program for which you want to evaluate servomotor duty.
- 10 When the program has finished executing, confirm the DUTY value of each axis.
- 11 **To change the motion group**, press F5, [UTIL], and select Group.
- 12 **To display help information**, press F5, [UTIL], and select Help. You will see a screen similar to the following.

STATUS/Axis		JOINT	100%
HELP	Arrows to scroll,	PREV	to exit
<p>If DUTY value is over 100% and you continues to run this program, motor may be overheated.</p> <p>Please use ACC or WAIT instruction to this program to reduce DUTY value less than 100%.</p>			
[TYPE]	GROUP		>

**9 PROGRAM AND
FILE MANIPULATION**

9

PROGRAM AND FILE MANIPULATION

Topics In This Chapter	Page
Storage Devices	You can store programs and files on three kinds of devices: controller memory, floppy disks, and the IBM PC or compatible personal computer. You can also print files to a serial printer. 9-2 <ul style="list-style-type: none">● Setting Up a Port 9-3● Connecting a Disk Drive to the Controller 9-8● Using a Memory Card Interface 9-11● Setting the Default Device 9-17● Formatting Disks 9-19
Manipulating Programs	A program is a series of robot commands that tells the robot and other equipment how to move and what to do to perform an application 9-27 <ul style="list-style-type: none">● Selecting Programs on the SELECT Menu 9-28● Saving Programs to Disk 9-29● Loading Programs from Disk 9-31● Copying Programs within the SELECT Menu 9-32● Deleting Programs from the SELECT Menu 9-34● Printing 9-35
Manipulating Files	Files are stored on disks: Floppy disk or personal computers. You can display a directory of files on a disk. You can also load files to be executed from a disk to controller memory, and display the contents of all text (ASCII) files. In addition, you can back up program and system files, copy files to a disk and delete files from a disk. 9-39 <ul style="list-style-type: none">● Generating a Directory of Files 9-41● Loading and Restoring Files from Disk to Controller Memory 9-44● Backing Up Program and System Files 9-53● Displaying Text (ASCII) Files 9-58● Deleting Files from a Disk 9-61● Saving Files 9-63● Checking and purging file memory 9-65
Controller Backup and Restore	Controller backup and restore allows an R-J2 controller to back up and restore controller memory: 9-66 <ul style="list-style-type: none">● Backing up a controller 9-66● Restoring a controller 9-72

A program is a series of robot commands that tell the robot and other equipment how to move and what to do to perform an application. A file is a unit in which the HandlingTool robot system stores information. Programs and files are manipulated and stored on storage devices.

9.1 STORAGE DEVICES

Three kinds of storage devices can be used to store programs and files:

- Controller memory
- Floppy disks
- IBM PC or compatible personal computer

This section describes how to set up storage devices for use. Depending on the storage device, this can include

- Setting up a port on the controller
- Connecting the device to the controller
- Formatting a disk

After you have set up the device(s) you will use, you must specify which device you want to use before you use it. This section contains instructions for selecting the default device.

Controller Memory

Controller memory is composed of FLASH ROM (non-volatile memory) and CMOS RAM which is battery-backed Random Access Memory that is located inside the controller. Programs are automatically stored in CMOS RAM when you write a program.

CMOS RAM is non-volatile memory. This means that all data in CMOS RAM, including programs, remains in CMOS RAM even after you turn off and turn on the controller. After your system is installed, you do not need to perform any setup operations for controller memory.

NOTE Volatile means the memory is lost when power is disconnected. Non-volatile memory does not require battery power to retain.

CAUTION

Data in CMOS RAM can be lost if the battery is removed or loses its charge, or if new system software is loaded on the controller. To prevent loss of data back up, or copy, all files for permanent storage.

Floppy Disks

Disk drives can be used to format magnetic floppy disks and copy or transfer files from the controller to disk. Types of disk drives include

- FANUC Handy File
- FANUC FLOPPY CASSETTE ADAPTER
- PS-100 Disk Drive – for 3.5” low density disks
- PS-200 Disk Drive – for 5.25” low density disks

If you use a disk drive, you connect it to the RS-232-C port on the controller. During file manipulations, the disk drive connected to the RS-232-C port is referred to as “P2:” on the FILE menu.

To set up a floppy disk drive for program and file manipulation, you must connect the disk drive to the controller and format a floppy disk. Procedure 9-5 describes how to format a floppy disk.

Personal Computer

An IBM PC or compatible personal computer (PC) can be used to store files off-line. You can use OLPC, the FANUC Robotics off-line programming software for the PC, to store files on a magnetic disk.

To set up a personal computer for program and file manipulation, you must set up the port on the controller to which it is connected and connect the personal computer to the controller. Section 9.1.1 describes how to set up a controller port.

Memory Card

The memory card device (MC:) is a 2 MB CMOS memory card. The memory card requires a memory card interface in the power supply unit of the CPU rack in the controller.

The memory card can be formatted and used as an MS-DOS file system. It can be read from and written to on the controller and an IBM PC equipped with the proper hardware and software. If the memory card is used as an MS-DOS file system, it should be formatted only on the R-J2 controller. Refer to Section 9.1.3 for information on installing and setting up a memory card.

**9.1.1
Setting Up a Port**

Setting up a port means initializing controller ports to use specific devices, such as the CRT/KB, printers, and disk drives. Initializing ports involves setting up specific information for a port based on the kind of device that will connect to the port.

The R-J2 controller supports two standard ports and two optional ports. Several different kinds of devices can be connected to these ports. Figure 9-1 shows the location of the standard and optional ports.

Standard Ports

There are two standard ports (P1 and P2) and two optional ports (P3 and P4). Table 9-1 lists the ports. You can set up ports P2 through P4. You cannot set up P1 (the teach pendant port).

Table 9-1. Ports, P1 – P4

Port	Item Name on Screen	Kind of Port	Use	Default Device
P1	Teach Pendant	RS-422	Teach pendant	Teach pendant
P2	RS-232-C	RS-232-C	Any device, such as a printer or disk drive	Handy File
P3	PORT 2	RS-232-C RS-422	Any device, such as a printer or disk drive	Printer
P4	JD17	RS-232-C RS-422	Any device, such as a printer or disk drive	No Use

Devices

You can modify the default communications settings for each port except port 1, which is dedicated to the teach pendant (TP). Table 9-2 lists the default settings for each kind of device you can connect to a port.

Table 9-2. Default Communications Settings for Devices

Device	Speed (baud)	Parity Bit	Stop Bit	Timeout Value (sec)
Handy file*	9600	None	2 bit	0
FANUC Floppy*	9600	None	2 bit	0
PS-100/200 floppy disk	9600	None	1 bit	0
Printer**	4800	None	1 bit	0
Sensor*	4800	Odd	1 bit	0
Host Comm.*	4800	Odd	1 bit	0
KCL/CRT	9600	None	1 bit	0
Debug console	9600	None	1 bit	0
Factory Terminal	9600	None	1 bit	0
TP Demo Device	9600	None	1 bit	0
No Use	9600	None	1 bit	0

* You can adjust these settings; however, if you do, they might not function as intended because they are connected to an external device.

** You can use only a serial printer.

Interfaces: RS-232-C and RS-422

On the SETUP Port screen, you can choose one of the following serial communications interfaces: RS-232-C or RS-422. The RS-422 interface is supported on port 2 and port 3. It is not supported on port 1.

RS-232-C Interface

- RS-232-C is available on port 1 (P2:), port 2 (P3:), and port 3 (P4:).
- The maximum cable length is approximately 50 feet (15 meters).

RS-422 Interface

- RS-422 is available on port 2 (P3:) and port 3 (P4:), but not on port 1 (P2:).
- The maximum cable length is approximately 50 meters.
- RS-422 provides more noise rejection
- RS-422 is useful in arc welding systems, because the data transfer function or sensor interface fails sometimes due to electrical noise.
- The electrical signal of RS-422 is different from the RS-232-C signal. If you need to connect between a robot controller and personal computer, you will need a converter, because normally a personal computer does not support the RS-422 interface.

Connector Pin Configuration

Refer to Table 9-3 for the pin configuration of the P3 port DB-25 connector. Refer to Table 9-4 for the pin configuration of the P4 port JD-17 connector (located on the Main CPU).

Table 9-3. Pin Configuration of the P3 Port DB-25 Connector

Pin	Signal
14	Tx
15	*Tx
16	Rx
17	*Rx

Table 9-4. Pin Configuration of the P4 Port JD-17 Connector

Pin	Signal
7	Rx
8	*Rx
17	Tx
18	*Tx

Use Procedure 9-1 to set up a port.

Procedure 9-1 Setting Up a Port

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select Port Init. You will see a screen similar to the following.

SETUP Port Init			JOINT	10 %
				1/3
	Connector	Port	Comment	
1	RS-232-C	P2:	[Handy File]	
2	PORT 2	P3:	[Printer]	
3	JD17 Main	PCBP4:	[No use]	
	[TYPE]		DETAIL	

- 5 Move the cursor to the port you want to set up and press F3, DETAIL. You will see a screen similar to the following.

SETUP Port Init			JOINT	10 %
RS-232-C	P2:			1/6
1	Device		[Handy File]	
2	Speed(Baud rate)		[9600]	
3	Parity bit		[None]	
4	Stop bit		[2bit]	
5	Time out value(sec)		[0]	
6	Interface		[RS-232-C]	
	[TYPE]		LIST [CHOICE]	

- 6 Select each item and enter the appropriate value using the [CHOICE] key.

NOTE To indicate that you are not using a port, set the port to **No use**.

- 7 A device cannot be assigned to two ports. Set one port to **No use** and the other to the appropriate device.
- 8 Perform a cold start of the controller to implement the changes to the Port Init screen:
 - a If the controller is turned ON, turn it OFF.
 - b On the teach pendant, press and hold the PREV and NEXT keys.
 - c While still pressing PREV and NEXT on the teach pendant, press the ON button on the operator panel or operator box.
 - d After the BMON> prompt appears on the teach pendant screen, release the PREV and NEXT keys.
 - e Press F1, COLD, and press ENTER.
 - f Press F5, START, and press ENTER.

BMON>

BMON> COLD

BMON> START

9.1.2 Connecting a Disk Drive to the Controller

The disk drives connect to the P2 controller port. The P2 controller port is an RS-232-C interface. The following disk drives are available:

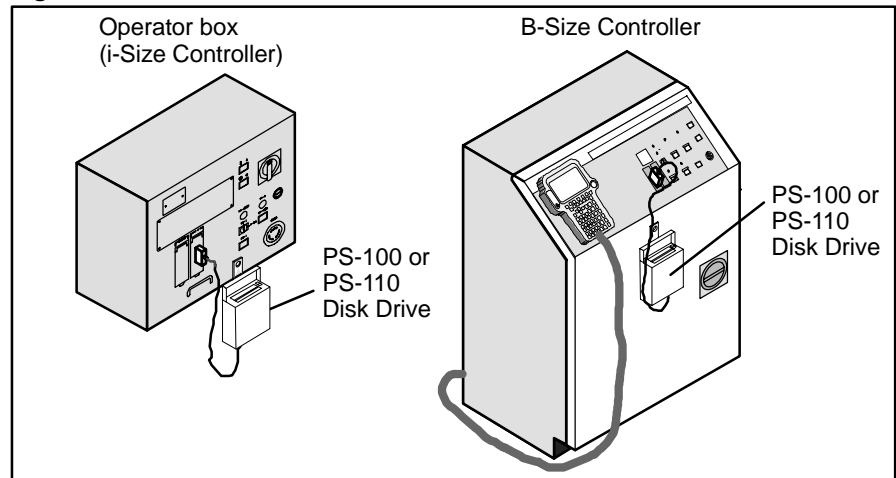
- Handy File
- FLOPPY CASSETTE ADAPTER
- PS-100
- PS-110
- PS-200

Use Procedure 9-2 to use a floppy disk and disk drive.

PS-100 Disk Drive

The PS-100 or PS-110 disk drive, shown in Figure 9-2, is used with 3 1/2 inch 720K double-density disks.

Figure 9-2. PS-100 or PS-110 Connected to the i-Size and B-Size Controller

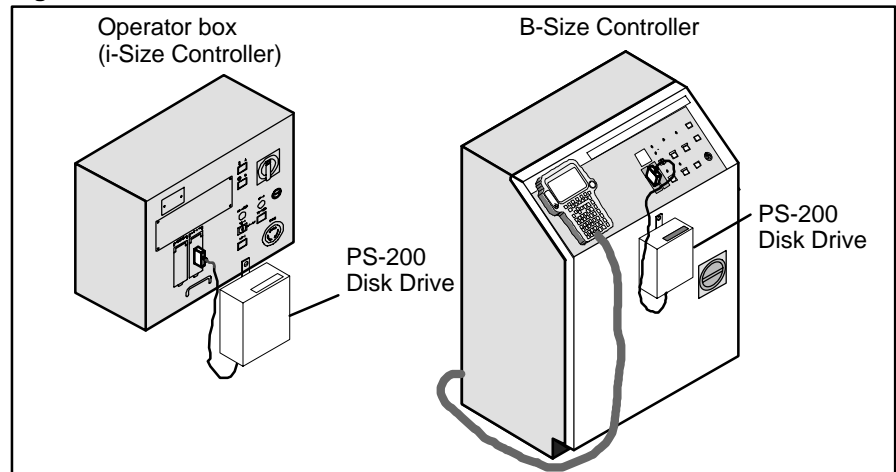


PS-200 Disk Drive

The PS-200 disk drive, shown in Figure 9-3, is used with 5 1/4 inch disks, including

- 360K double-density disks

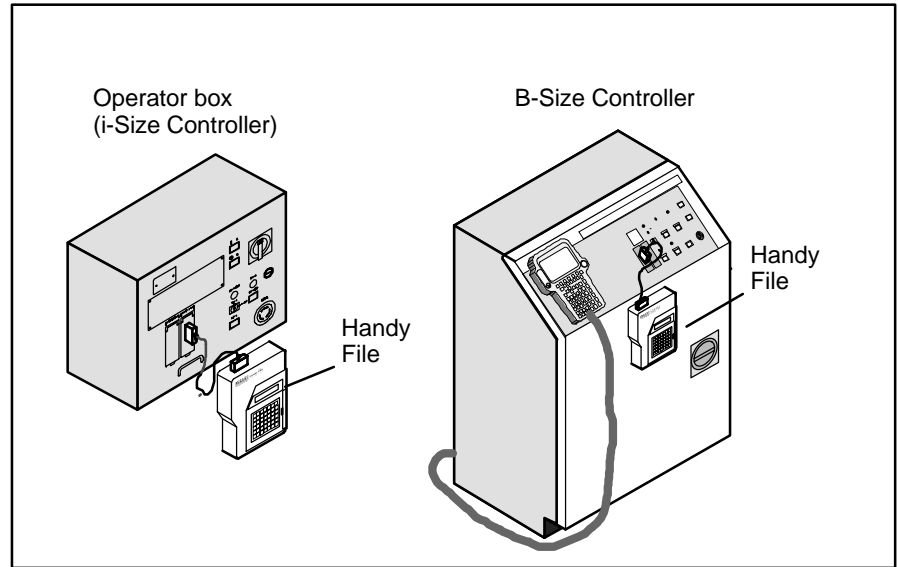
Figure 9-3. PS-200 Connected to the i-Size and B-Size Controller



Handy File
(not available in North American HandlingTool)

The Handy File, shown in Figure 9-4, is used with 3.5 inch 1.44M high density disks.

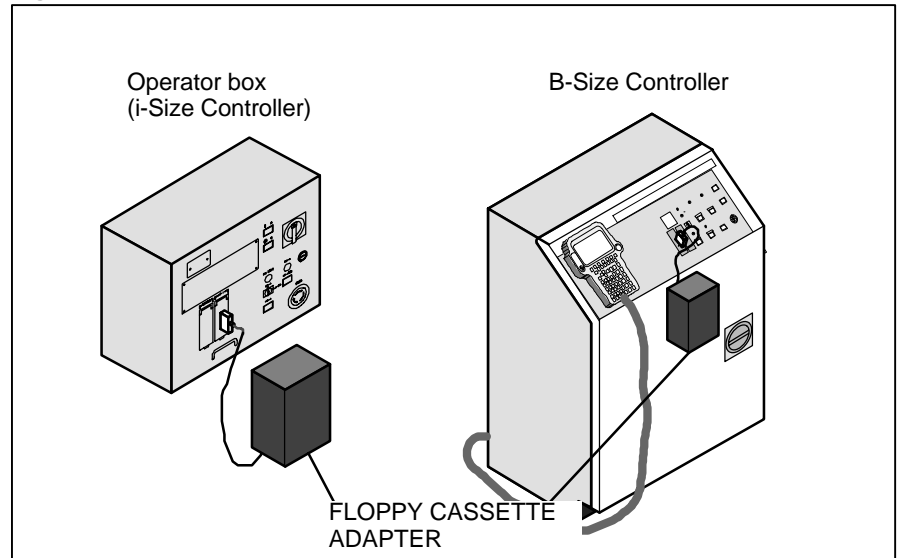
Figure 9-4. Handy File Connected to the Controller



FANUC Floppy
(not available in North American HandlingTool)

The FLOPPY CASSETTE ADAPTER, shown in Figure 9-5, is used with 3.5 inch 1.44M high density disks.

Figure 9-5. FLOPPY CASSETTE ADAPTER Connected to the Controller



Procedure 9-2 Using a Floppy Disk and Disk Drive

 CAUTION

If devices such as a printer, floppy disk drive, or vision system are connected to the controller, always turn on the robot first, then turn on these devices; otherwise, equipment could be damaged.

- Step**
- 1** Connect the RS-232-C cable from the disk drive to the P2 controller port.
 - 2** Turn on the disk drive.
 - **For the PS-100 or PS-110**, turn on the power switch located under the disk drive cover. The LED next to the power switch will turn on.
 - **For the PS-200**, connect the disk drive to 110 VAC power and turn on the power switch located in the rear of the disk drive.
 - 3** Hold the disk with the label toward you and insert it into the disk drive.
 - 4** Format the disk if necessary, using Procedure 9-5 .

9.1.3 Using a Memory Card Interface

The memory card interface can connect either to the ER-2 board or directly to the R-J2 controller backplane.

Figure 9-6 shows the memory card interface module inserted into the controller. Use Procedure 9-3 to install a memory card in a controller.

NOTE Loading from a memory card is not available as a standard product.

CAUTION

Be sure that the version of Main CPU BootROM you have is version 4.20 or later. If not, DO NOT load the optional A-B RIO software from memory card while the memory card interface is in the ER-2 printed circuit board. Otherwise, you could erase all the information on the memory card and destroy the ER-2 printed circuit board. Instead, if the BootROM version is not 4.20 or later, load the A-B RIO software using the FANUC Robotics-supplied disks.

Figure 9-6. Memory Card Interface and Memory Card Connected to the i-size Controller

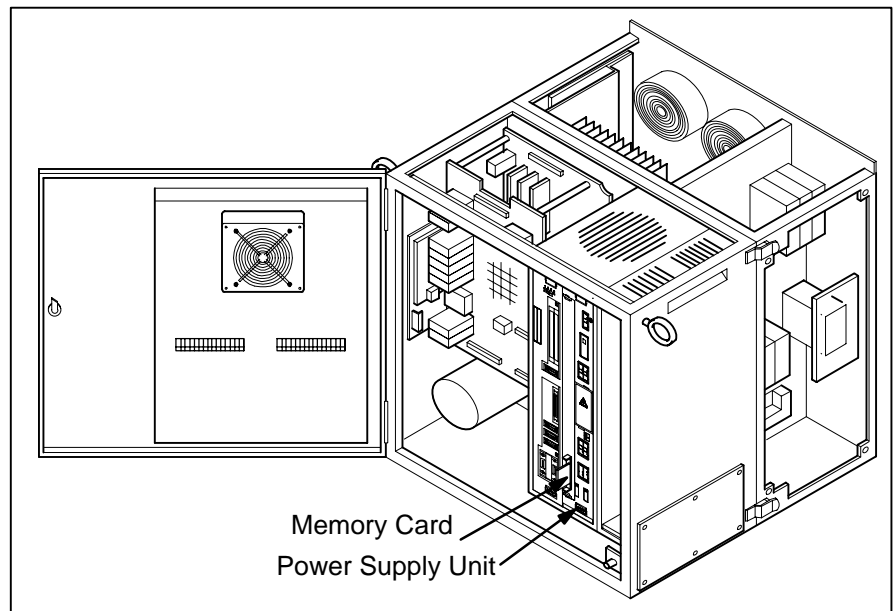
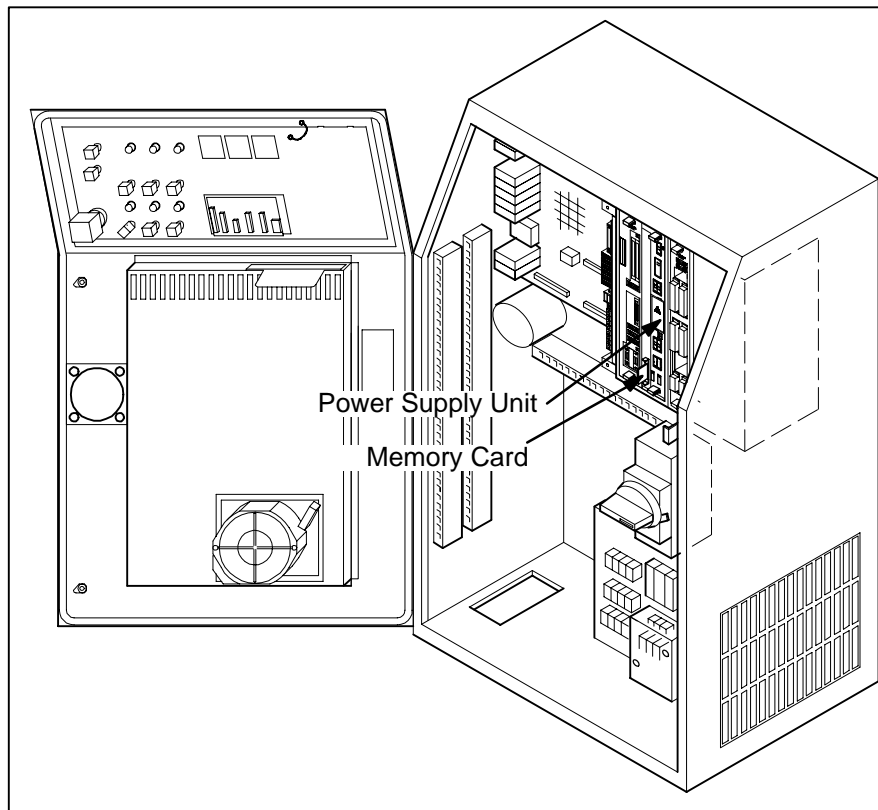


Figure 9-7. Memory Card Interface and Memory Card Connected to the B-Size Controller



Procedure 9-3 Using the Memory Card Interface

NOTE Loading from a memory card is not available as a standard product.

CAUTION

If devices such as a printer, floppy disk drive, or vision system are connected to the controller, always turn on the robot first, then turn on these devices; otherwise, equipment could be damaged.

Condition

- The controller is turned off.
- You have a memory card interface module and memory card that contains the software you want to load.
- You are using memory cards that are based on one of the following standards:
 - JEIDA “IC Memory Card Guideline Version 4.0”
 - PCMCIA “PC Card Standard R. 2.0”
 - 2 MB SRAM card (no Flash ROM cards)

⚠ WARNING

Disconnect electrical power from the controller before you remove or replace components, or you could be injured seriously.

⚠ CAUTION

Use anti-static devices and observe anti-static safety precautions when handling any electronic material, otherwise you could damage the equipment.

- You are wearing a wrist strap to prevent static discharge to the C-MOS circuits.

Step

1 Disconnect electrical power from the controller.

- **If your controller is equipped with a disconnect handle** at the upper right front corner, pull it to the OFF (down) position.

OR

- **If your controller is equipped with a circuit breaker handle**, turn the handle to the OFF (open) position.

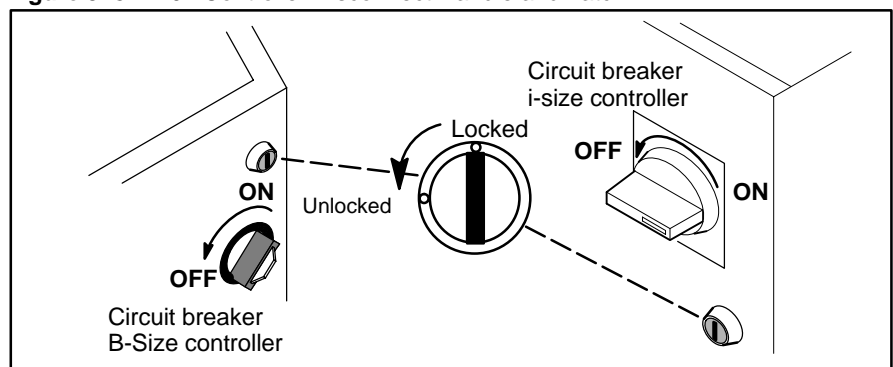
See Figure 9-8.

⚠ WARNING

When the disconnect or circuit breaker handle is OFF, power is still present inside the controller. You must unplug the controller from the electrical outlet to remove all power from the controller.

- 2** Open the front door of the controller using a flat-tip screwdriver to turn the latch. See Figure 9-8.

Figure 9-8. R-J2 Controller Disconnect Handle and Latch



- 3 Identify the memory card interface module location inside the controller. See Figure 9-9 and Figure 9-10.

Figure 9-9. Memory Card Interface Location on an i-size Controller

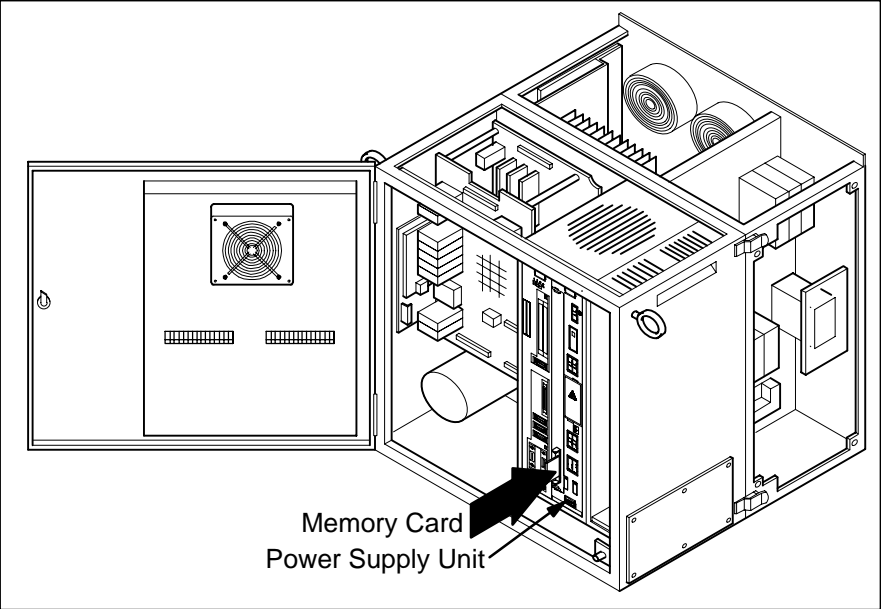
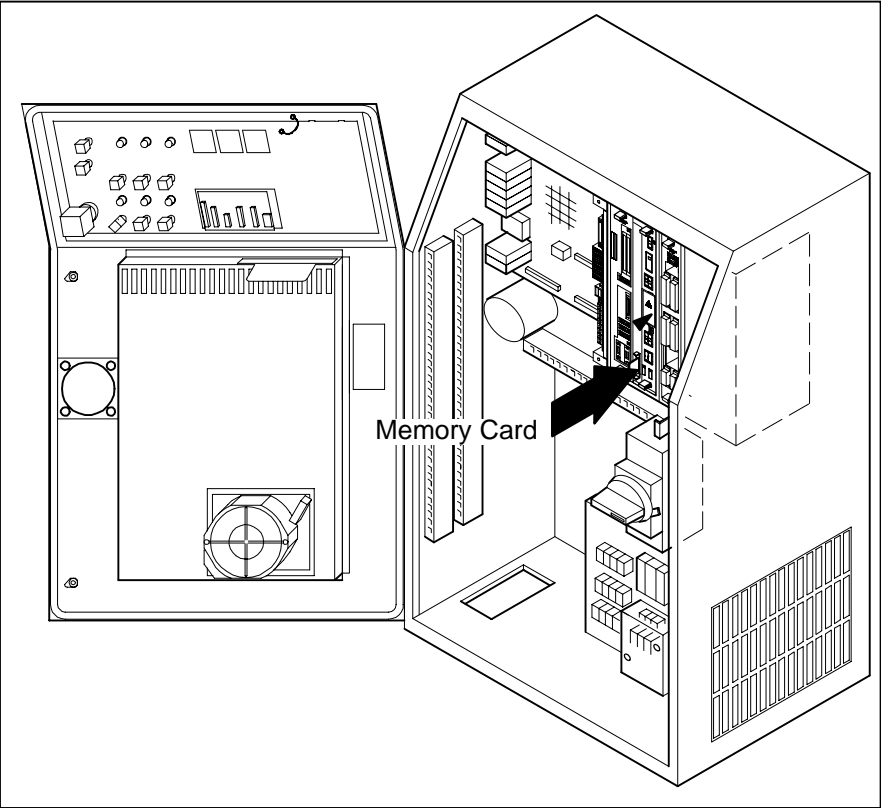
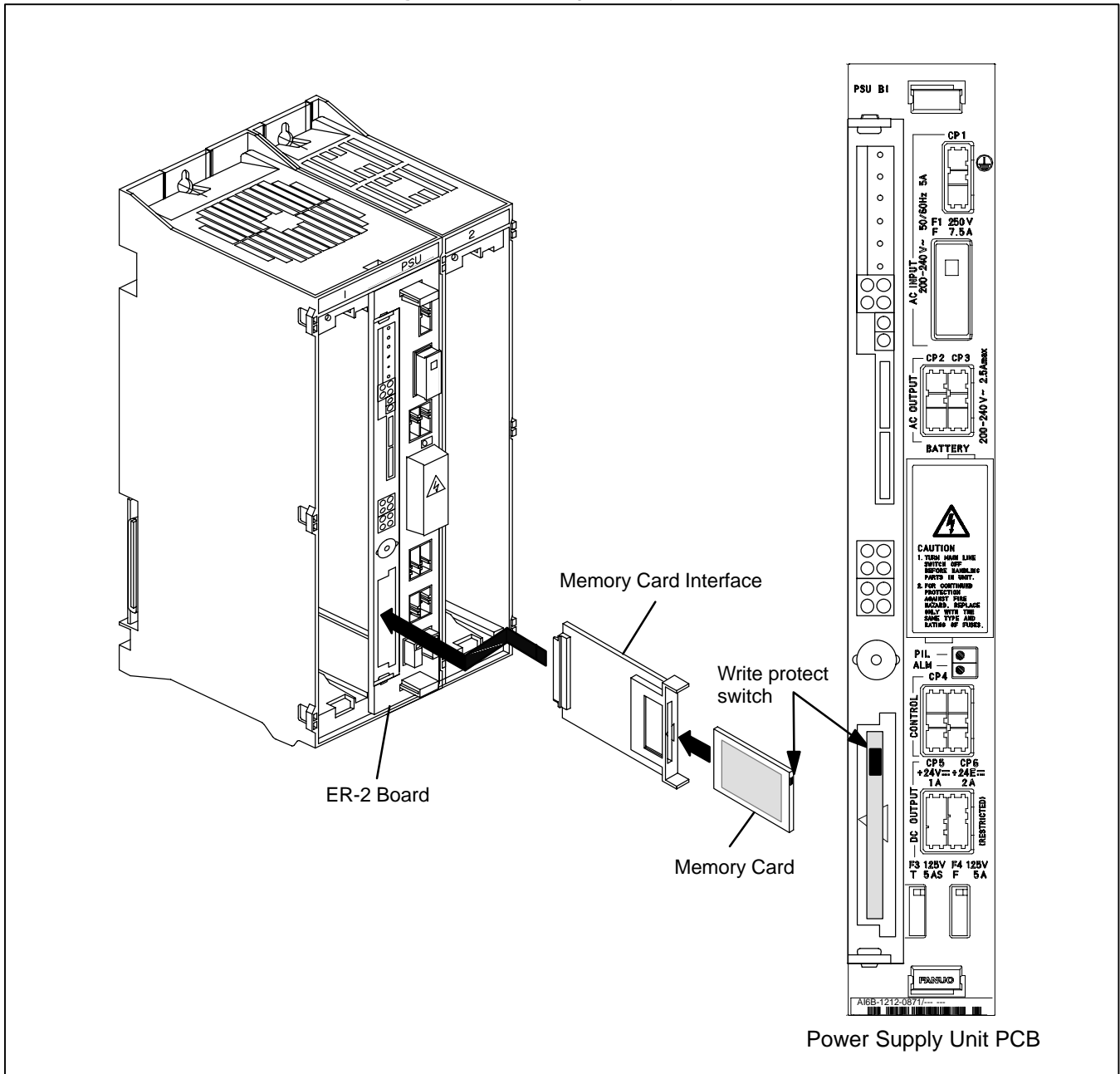


Figure 9-10. Memory Card Interface Location on a B-Size Controller



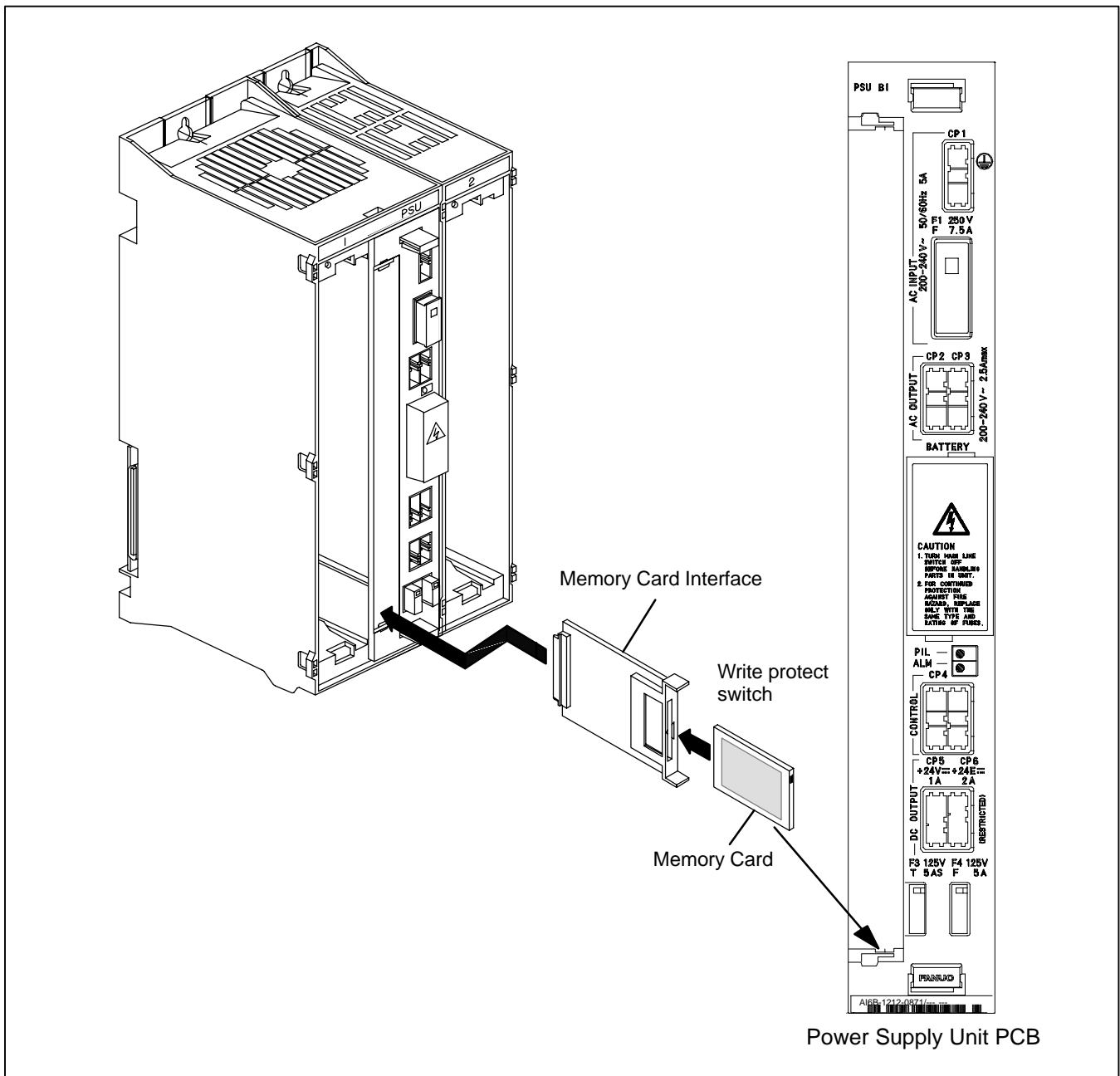
- 4 To insert a memory card, hold the disk with the label facing the left and the write protect switch on the top.**
- If you have an ER-2 printed circuit board, see Figure 9-11.

Figure 9-11. Inserting a Memory Card with the ER-2 Printed Circuit Board



- If you do not have an ER-2 printed circuit board, insert the memory card interface in the module located in the 1/2 slot next to the power supply unit. The interface will only fit in the left slot. See Figure 9-12.

Figure 9-12. Inserting a Memory Card without an ER-2 Printed Circuit Board



NOTE Loading from a memory card is not available as a standard product.

⚠ CAUTION

Do not close the i-size controller door when the memory card is in the interface. Otherwise, you could damage the memory card.

⚠ WARNING

Lethal voltage is present in the controller WHENEVER IT IS CONNECTED to a power source. Be extremely careful to avoid electrical shock when the controller door is open.

Turning the disconnect or circuit breaker to the OFF position removes power from the output side of the device only. High voltage is always present at the input side whenever the controller is connected to a power source.

- 5** If you are using a B-size controller, close and latch the controller door.

9.1.4 Setting the Default Device

Setting the default device specifies which device to use when manipulating programs and files. You must set the default device before you can perform any program or file manipulations, including formatting a disk. You can set the default device to

- Serial floppy disk – A serial floppy disk drive connected to the P2 port of the controller, such as the PS-100, PS-110, PS-200, Handy File or FLOPPY CASSETTE ADAPTER.
- Serial printer (text only) – A serial printer connected to the P2 port of the controller
- Client tag (C1: – C8:) – used if the FTP option is installed. The client devices displayed are the client devices that have been defined and started.
- Memory card (MC:) – displayed if the memory card interface is installed.

After you set the default device, the device will remain the default until you change it or until an init start is performed.

Use Procedure 9-4 to set the default device.

Procedure 9-4 Setting the Default Device

⚠ CAUTION

Before you connect the floppy disk to the controller, turn on the controller, then connect and turn on the floppy disk; otherwise, equipment could be damaged.

- Condition**
- If you are setting the default device to P2, the PS-100, PS-200, Handy File, FLOPPY CASSETTE ADAPTER, printer, or other device is connected to the P2 port on the controller and is turned on.

- Step**
- 1 Press MENUS.
 - 2 Select FILE. You will see a screen similar to the following.

```

FILE                                JOINT 10%
FLPY:\*.*                               1/16
1 * * (all files)
 2 *      KL (all KAREL source)
 3 *      CF (all command files)
 4 *      TX (all text files)
 5 *      LS (all KAREL listings)
 6 *      DT (all KAREL data files)
 7 *      PC (all KAREL p-code)
 8 *      TP (all TP programs)
 9 *      MN (all MN programs)
10 *      VR (all variable files)
Press DIR to generate directory

11 *      SV (all system files)
12 *      IO (I/O config data)
13 *      DF (all DEFAULT files)
14 *      ML (all part model files)
15 *      BMP (all bit-map images)
16 [you enter]
Press DIR to generate directory
[TYPE] [DIR]  LOAD  [BACKUP] [UTIL] >

DELETE COPY DISPLAY >

```

1 Set Device
2 Format

- 3 Press F5, [UTIL].
- 4 Select Set Device.

```

1 Floppy disk
2 Serial Printer
3 FROM Disk (FR:)
4 FTP (Cl:)
5 Memory Card (MC:)

```

5 Move the cursor to the device you want to set as the default and press ENTER to select it. See the following screen for an example.

```

FILE                                JOINT 10%
FLPY:\*.*

[TYPE]  [DIR]  LOAD  [BACKUP]  [UTIL] >

```

The default device is now set. The default device name is displayed in the FILE menu, under the word "FILE."

9.1.5 Formatting Disks

You must format the floppy disk only before you use them for the first time. Use Procedure 9-5 to format a floppy disk.



CAUTION

Formatting deletes all the files on a disk. Do not format a disk that contains files you want to keep. Otherwise, you will delete all of the files on the disk.

Procedure 9-5 Formatting a Floppy Disk

- | | |
|------------------|--|
| Condition | <ul style="list-style-type: none"> ■ The disk drive is connected to the controller. ■ The default device is set to P2. Refer to Procedure 9-4 . ■ The floppy disk is not write protected. |
| Step | <ol style="list-style-type: none"> 1 Turn on the disk drive. 2 Hold the disk to be formatted with the label toward you and insert it into the disk drive. 3 Press MENUS. |

1 Set Device
2 Format

- 4 Select FILE.
- 5 Press F5, [UTIL].
- 6 Select Format. You will see a screen similar to the following.

```

File Format                                JOINT 10%
FLPY:\*.*
          Formatting FLPY:

***** WARNING *****
          ANY DATA ON THE DISK WILL BE LOST!
Insert the disk to be formatted into the
          disk drive

Format disk?                                YES    NO

```

- 7 Format the floppy disk:
 - **If you do not want to format the floppy disk**, press F5, NO.
 - **To format the floppy disk**, press F4, YES. You will see a screen similar to the following.
- 8 Use the appropriate function keys and numeric keys to type a volume label, such as disk1, and press ENTER.

Formatting floppy disk: disk1

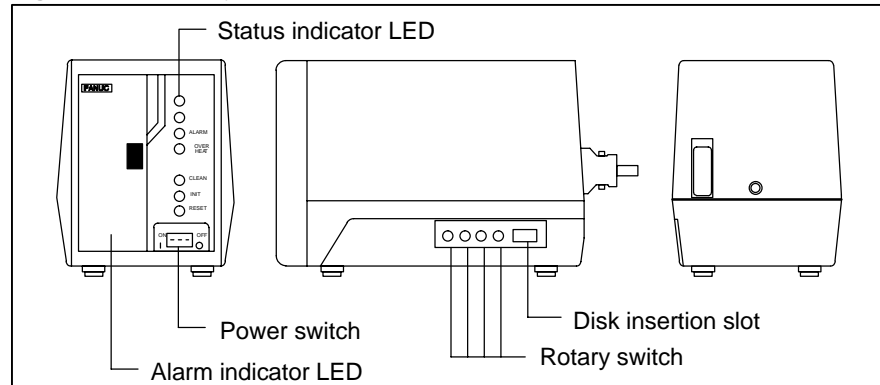
Formatting disks takes a few minutes. When the formatting is complete the teach pendant FILE menu will be displayed.

FLOPPY CASSETTE ADAPTER

(not available in North American Handling Tool)

The Floppy Cassette adapter is an external memory unit connected to the R-J2 Mate controller to save files stored in the internal memory of the controller to a floppy disk or read files from a floppy disk. For detailed information about the Floppy Cassette adapter (A16B-0150-B001), refer to the *FANUC Floppy Cassette Adapter Operator's Manual*, (B-66040E). See Figure 9-13.

Figure 9-13. Floppy Cassette Adapter



Rotary Switch Setting

For port setting on the Floppy Cassette adapter, rotary switches 1 to 4 on the side panel are used. The standard settings for connection with the R-J2 controller are "3, 1, 0, 0." Refer to Table 9-5.

Table 9-5. Port Setting on Floppy Cassette Adapter

	Speed	Stop bit	Parity bit	Number of files	Data code
Standard setting	9600	2 bit	None	71	ISO
Switch		(1) 3	(2) 1	(3) 0 (4) 0	

Status Indicator LEDs

The status indicator LEDs on the Floppy Cassette adapter indicate operation status. See Figure 9-14 and Table 9-6.

Figure 9-14. Status indicator LEDs

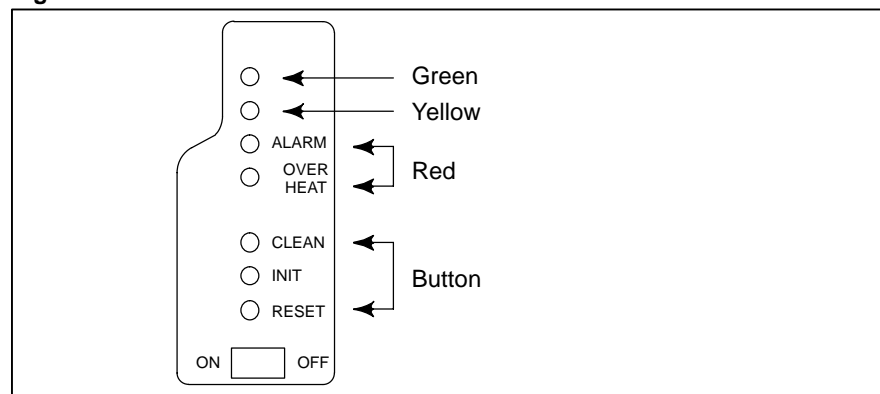


Table 9-6. Status Indicator LEDs and Switches

Green	Yellow	Status
Blinking alternately		No floppy disk is inserted, or the door is not closed.
On	On(*1)	Ready (with write protection not applied)
On	Blinking	The floppy disk is being formatted.
Blinking	On	The floppy disk is being cleaned.
On	Blinking	Data is being written.
Blinking	On(*1)	Data is being read.
Blinking simultaneously		A file is being deleted.

NOTE *1 Turned off when the disk is write protected.

Button	Status
CLEAN	Used to clean the head
INIT	Used to format a floppy disk
RESET	Used to release an alarm

Procedure 9-6 Operating the Floppy Cassette Adapter

- Step**
- 1** Connect the Floppy Cassette adapter to the controller.
 - 2** Turn on the power to the Floppy Cassette adapter. The green LED and yellow LED blink alternately.
 - 3** Insert a floppy disk, then close the door. The green LED and yellow LED light to indicate that the Floppy Cassette adapter is ready for operation. If the disk is write protected, the yellow LED does not light.

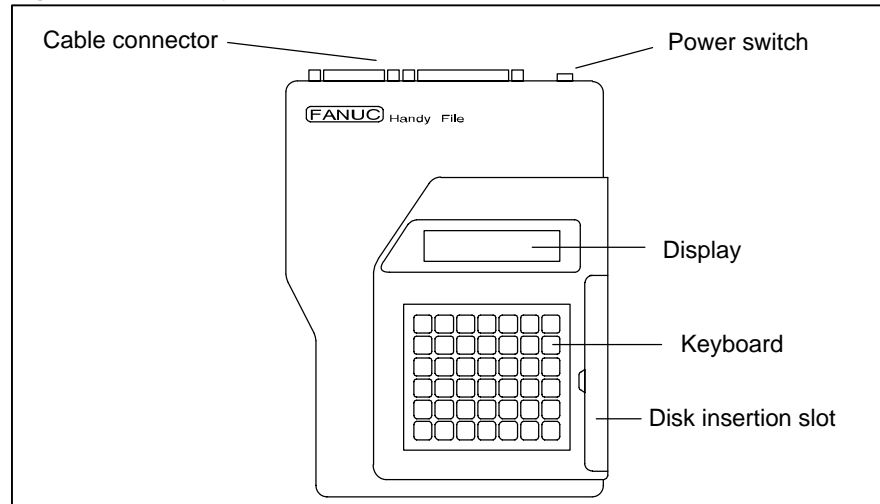
NOTE The Floppy Cassette adapter cannot be used if the door is not closed.

- 4** To format the floppy disk, press the RESET button while holding down the INIT button.
- 5** To clean the head, press the RESET button while holding down the CLEAN button.
- 6** If an alarm is issued, press the RESET button.

Handy File
(not available in North American HandlingTool)

The Handy File is an external memory unit connected to the R-J2 controller to save files stored in the internal memory of the controller to a floppy disk or read files from a floppy disk. For detailed information about the Handy File (A16B-0159-B002), refer to the *FANUC Handy File Operator's Manual*, (B-61834E). See Figure 9-15.

Figure 9-15. Handy File



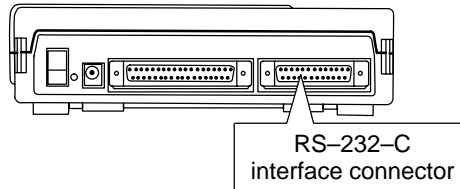
The setting screen is used for port setting on the Handy File. Refer to Table 9-7.

Table 9-7. Port Setting for Handy File

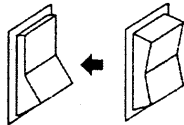
Item	Standard setting	
Protocol	Others	
Receive start/stop code	DC1/DC3(X11/X93)	
Send start/stop code	DC1/DC3(X11/X93)	
Receive data beginning/end code	DC2/DC4(X12/X14)	
Send data beginning/end code	DC2/DC4(X12/X14)	
Speed	9600 baud	
Stop bit	2 bits	
Parity bit	None	
Data code	Receive	ISO/EIA
	Send	ISO
Channel	RS-232-C	
Subprogram	None	

Procedure 9-7 Operating a Handy File

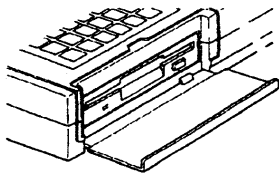
Step 1 Connect the Handy File to the controller.



2 Turn on the power to the Handy File.



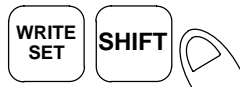
3 Insert a floppy disk, then close the door. The Handy File is now ready for operation.



No file
Ready

4 Port setting

The setting menu is used for port setting. Press the WRITE/SET key while holding down the SHIFT key. The setting menu appears.



Select setting item
#1 : Input/Output



Select setting item
#2 : Protocol

While on the setting menu, switch between menu items with the up and down arrow keys. To select an item, press the ENTER key.

5 Select "#2: Protocol" for protocol setting.



Protocol :
#3 : Other



Receive Start/Stop
#1 : DC1/DC3(X11/X93)

While on the setting menu, switch between menu items with the up and down arrow keys. To select an item, press the ENTER key.

- 6** Upon completion of protocol setting, press the END key.



```
Select setting item
#3 : Baud rate
```

- 7** When all menu items have been set, press the END key.



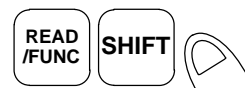
```
No file
Ready
```

8 Formatting the floppy disk

When the floppy disk is not formatted, a message is displayed.

```
! FD format error
Initialize FD.
>FUNC: SELECT FUNCTI.
```

- 9** The miscellaneous menu is used to format the floppy disk. To display the miscellaneous menu, press the READ/FUNC key while holding down the SHIFT key.



```
Select function
#1 : Initialize FD
```

- 10** Select “#1: Initialize FD” to format the floppy disk.

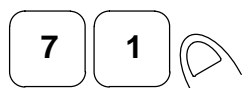


```
Select format of FD
#1 : 2HD, 1.02MB FANUC
```

- 11** Select a format. For this example, select “FANUC.”

```
Set number of file
> Maximum =
```

- 12** Set a maximum number of files. For this example, enter “71.”



```
Set number of file
> Maximum = 71
```



```
Initialize FD :
Press START key
```

- 13** Press the START key to start the formatting of the floppy disk.



```
Initialize FD :  
> Executing
```

```
Initialize FD :  
> Complete
```

- 14** When you are finished formatting the floppy disk, press the END key.



```
Select function  
#1 : Initialize FD
```

- 15** To exit the miscellaneous menu, press the END key.



```
No file  
Ready
```

- 16** **Cleaning the head**

The miscellaneous menu is used to clean the head. Select “#2: Cleaning” to clean the head.

```
Select function  
#2 : Cleaning
```

- 17** Press the START key to start cleaning the head. When you finished cleaning the head, press the END key.

9.2 MANIPULATING PROGRAMS

As programs are created they are stored automatically on controller memory. A list of all programs stored on controller memory is displayed on the SELECT menu. See the following screen for an example.

Select		JOINT 10%	
		513712 bytes free	1/4
No.	Program name	Comment	
1	MAIN_A	[]
2	MAIN_B	[]
3	SUB_A	[]
4	SUB_B	[]
[TYPE] CREATE DELETE MONITOR [AFTER]>			
COPY DETAIL LOAD SAVE PRINT >			

Programs can be:

- Selected
- Saved to a disk
- Loaded from a disk
- Copied within the SELECT menu
- Deleted from the SELECT menu
- Monitored
- Printed

9.2.1 Selecting Programs on the SELECT Menu

You can select programs on the SELECT menu. Selecting a program chooses the program as the current program, for modifying, testing, or executing. Use Procedure 9-8 to select a program on the SELECT menu.

Procedure 9-8 Selecting a Program on the Select Menu

Step 1 Press SELECT. You will see a screen similar to the following.

Select		JOINT 10%	
No.	Program name	50983 BYTES FREE	1/6 Comment
1	SUB1	[]
2	MAIN25	[]
3	PRG7	[]
4	JOB0001		
5	PROC0010		
6	TEST		

[TYPE]	CREATE	DELETE	MONITOR	[ATTR]	>
--------	--------	--------	---------	--------	---

COPY	DETAIL	LOAD	SAVE	PRINT	>
------	--------	------	------	-------	---

2 Press F1, [TYPE].

3 Select the list you want:

- **All** displays all programs.
- **TP Programs** displays all teach pendant programs.
- **Macro** displays all macro programs.

4 Select the name of the program you want and press ENTER.

Select		JOINT 10%	
No.	Program name	513712 bytes free	1/4 Comment
1	MAIN_A	[]
2	MAIN_B	[]
3	SUB_A	[]
4	SUB_B	[]

[TYPE]	CREATE	DELETE	MONITOR	[AFTER]	>
----------	--------	--------	---------	---------	---

COPY	DETAIL	LOAD	SAVE	PRINT	>
------	--------	------	------	-------	---

9.2.2 Saving Programs to Disk

Saving programs allows you to save a program and its relevant data to a disk. Refer to Section 9.1 for the kinds of disks available. Use Procedure 9-9 to save a program to a disk.

Procedure 9-9 Saving a Program to a Disk

- Condition**
- The default device is set. Refer to Procedure 9-4 .
 - **If you are saving programs to a serial floppy disk**, be sure it is connected to the controller P2 port, is turned on, and contains a formatted floppy disk.

CAUTION

Before you connect the floppy disk drive to the controller, turn on the controller, then connect and turn on the floppy disk drive; otherwise, equipment could be damaged.

- Step** 1 Press SELECT. You will see a screen similar to the following.

Select		JOINT 10%	
		513712 bytes free	1/4
No.	Program name		Comment
1	MAIN_A	[]
2	MAIN_B	[]
3	SUB_A	[]
4	SUB_B	[]
[TYPE] CREATE DELETE MONITOR [AFTER]>			
COPY DETAIL LOAD SAVE PRINT >			

- 2 Move the cursor to the program you want to save.

- 3 Press NEXT, > and then press F4, SAVE. You will see a screen similar to the following.

```

                                JOINT 10%
1 Words
2 Upper Case
3 lower Case
4 Options                      --Insert--
Select
--- Save Teach Pendant Program ---
Program Name [MAIN_A ]
Enter program name
PRG      MAIN      SUB      TEST
```

- 4 Type the program name to save and press ENTER.

NOTE Do not include the file extension.

The program will be saved to the default device as progname.tp regardless of its file extension on the controller. The SELECT menu will then be displayed.

9.2.3 Loading Programs from Disk

Loading programs allows you to load programs from a disk onto controller memory. A program must be loaded into controller memory and listed on the SELECT menu before it can be modified or executed. Use Procedure 9-10 to load programs.

Procedure 9-10 Loading a Program

CAUTION

Before you connect the floppy disk drive to the controller, turn on the controller, then connect and turn on the floppy disk drive; otherwise, equipment could be damaged.

Condition

- The default device is set. Refer to Procedure 9-4 .
- **If you are loading programs from a floppy disk drive**, be sure the disk drive is connected to the controller P2 port, is turned on, and the appropriate floppy disk is inserted into the disk drive.

Step

- 1 Press SELECT. You will see a screen similar to the following.

Select		JOINT 10%	
		513712 bytes free	1/4
No.	Program name	Comment	
1	MAIN_A	[]
2	MAIN_B	[]
3	SUB_A	[]
4	SUB_B	[]
[TYPE] CREATE DELETE MONITOR [AFTER]>			
COPY DETAIL LOAD SAVE PRINT >			

- 2 Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

- 3 Press NEXT, >, and then press F3, LOAD. You will see a screen similar to the following.

```

                                JOINT 10%
1 Words
2 Upper Case
3 lower Case
4 Options                      --Insert--
Select
--- Load Teach Pendant Program ---
Program Name [          ]
Enter program name
PRG      MAIN      SUB      TEST

```

- 4 Type the program name to load and press ENTER.

NOTE Do not include the file extension.

- 5 Load the selected program:
- If you do not want to load the selected program, press F2, NO.
 - If you want to load the selected program, press F1, YES.

The program you specified will be loaded from the default device onto controller memory. The SELECT menu will be displayed and the loaded program will appear on the menu.

9.2.4 Copying Programs Within the SELECT Menu

Programs can be copied within the SELECT menu. This means that both the original program and the copied program will be on controller memory. Use Procedure 9-11 to copy programs within the SELECT menu.



WARNING

Before copying a program with embedded macros from one controller to another, compare the Setup menu macro lists of the two controllers. Be sure that the list on the first controller matches the list on the second controller. If they are not identical, DO NOT copy the program; otherwise, unexpected results could occur.

Procedure 9-11 Copying a Program within the SELECT Menu

Step 1 Press SELECT. You will see a screen similar to the following.

Select		JOINT 10%	
		513712 bytes free	1/4
No.	Program name	Comment	
1	MAIN_A	[]
2	MAIN_B	[]
3	SUB_A	[]
4	SUB_B	[]
[TYPE] CREATE DELETE MONITOR [AFTER]>			
COPY DETAIL LOAD SAVE PRINT >			

2 Move the cursor to the program you want to copy.

3 Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

4 Press NEXT, >, and then press F1, COPY. You will see a screen similar to the following.

		JOINT 10%	
1	Words		
2	Upper Case		
3	lower Case		
4	Options	--Insert--	
Select			
--- Copy Teach Pendant Program ---			
From :		[MAIN_A]	
To :		[]	
		-- End --	
Press Enter for next item			
PRG	MAIN	SUB	TEST

5 Type the program name to which to copy the selected program. Press ENTER.

6 Copy the selected program:

- If you do not want to copy the selected program, press F5, NO.
- If you want to copy the selected program, press F4, YES.

The selected program will be copied. The SELECT menu will be displayed. If the copied program has a new name, it will be displayed in the SELECT menu.

Copy OK ?

YES NO

9.2.5 Deleting Programs from the SELECT Menu

If you no longer want to have a program loaded on controller memory (displayed on the SELECT menu) you can delete it. If you want to keep a copy of the program, save it to floppy disk before you delete it from the SELECT menu.

NOTE Deleting a program from controller memory does not delete it from a floppy disk.

Use Procedure 9-12 to delete a program from the SELECT menu. For information on deleting a program from a floppy disk, refer to Procedure 9-21.

Procedure 9-12 Deleting a Program from the SELECT Menu

Condition ■ The program you want to delete is listed on the SELECT menu.

Step 1 Press SELECT. You will see a screen similar to the following.

Select		JOINT 10%	
		513712 bytes free	1/4
No.	Program name	Comment	
1	MAIN_A	[]
2	MAIN_B	[]
3	SUB_A	[]
4	SUB_B	[]
[TYPE] CREATE DELETE MONITOR [AFTER]>			
COPY DETAIL LOAD SAVE PRINT >			

2 Move the cursor to the name of the program you want to delete.

3 Continuously press the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

4 Press NEXT, >, and then press F3, DELETE.

5 Delete the program:

- If you do not want to delete the selected program, press F5, NO.
- If you want to delete the selected program, press F4, YES.

Delete OK ?

YES NO

The program will be deleted from controller memory. The SELECT menu will be displayed and the deleted program will no longer be listed.

9.2.6 Printing

Programs and teach pendant screens can be printed to a serial printer or an ASCII file (optional feature). The printer must be properly connected and set up before you can print information from the controller.

Printer Requirements

The printer you use must meet the following requirements:

- The printer must be a serial printer. If you use a parallel printer, you will damage the controller and the printer.
- The printer must be connected to an RS-232-C port on the controller. Refer to Section 9.1.1 for information on setting up a port for a printer.
- The printer must be set up to use the RS-232-C port. Refer to the specifications for your printer for the proper communications settings.

Use Procedure 9-13 to print a program. Use Procedure 9-14 to print a teach pendant screen.

ASCII File Output

You can save the file program settings to an ASCII file as an optional feature.

If the selected device is set up as “Printer,” then the output is printed as ASCII text. If the device is set up as something other than “Printer,” then the output depends on the format of the device.

For example,

- For P2 set up as FLPY:, the output is a .LS file
- For RD:, the output is a .LS file
- For KCL, the output is displayed on the KCL screen

Refer to Table 9-8 for information on how a file will be output when you print it, under various conditions.

CAUTION

ASCII files can not be loaded on to the controller. To back up programs or settings, save the binary files using the file screen. Refer to 9.3.3.

Table 9-8. File Output Using PRINT

Operation	Output data	File name
Press F5, PRINT on the second page of the SELECT screen	The current program selected by the cursor	(program name).LS
Select PRINT FCTN menu on SELECT screen	The current program selected by the cursor	(program name).LS
Select PRINT on FCTN menu on EDIT screen	The editing program	(program name).LS
Select PRINT on FCTN menu on SYSTEM VARIABLE screen	System variables	SYSVAR.LS

Table 9-8. (Cont'd) File Output Using PRINT

Operation	Output data	File name
Select SCREEN PRINT in FCTN menu at the teach pendant	The current screen image	TPSCRN.LS
Select SCREEN PRINT on FCTN menu at the CRT	The current CRT screen image	CTSCRN.LS


ASCII Files (.LS)

You can print an ASCII file to a floppy disk or printer. When you save an ASCII file to an MS-DOS formatted floppy disk, you can read the file with an editor on a personal computer. You can also print the ASCII file using a printer connected to a personal computer.

NOTE You cannot load an ASCII file onto the controller.

Procedure 9-13 Printing a Program

- Condition**
- The printer is a serial printer.
 - The printer is connected to the P2 or P3 port and is set up properly to use that port. Refer to Procedure 9-1 .

 **WARNING**
Make sure the printer is a serial printer before you continue; otherwise, you could damage the controller and the printer.

- Step**
- 1 Turn on the printer if you have not already done so.
 - 2 Set the default device to serial printer:
 - a Press MENUS.
 - b Select FILE.
 - c Press F1, [TYPE].
 - d Select File.
 - e Press F5, [UTIL].
 - f Select Set Device.
 - g Move the cursor to Serial Printer and press ENTER.

3 Press SELECT. You will see a screen similar to the following.

Select		JOINT 10%	
		513712 bytes free	1/4
No.	Program name	[Comment
1	MAIN_A	[]
2	MAIN_B	[]
3	SUB_A	[]
4	SUB_B	[]
[TYPE] CREATE DELETE MONITOR [AFTER]>			
COPY DETAIL LOAD SAVE PRINT >			

4 Select the name of the program you want to print.

5 Press NEXT, > and then F5, PRINT. You will see a screen similar to the following.

		JOINT 10%	
1	Words		
2	Upper Case		
3	lower Case		
4	Options	--Insert--	
Select			
--- Print Teach Pendant Program ---			
Program Name [MAIN_A]			
Enter program name			
PRG	MAIN	SUB	TEST

6 Type the name of the program you want to print and press ENTER. The program will be printed.

NOTE To pause printing, press PREV key.

Procedure 9-14 Printing a Teach Pendant Screen

- Condition**
- The printer is a serial printer.
 - The printer is connected to the port and is set up properly to use that port. Refer to Procedure 9-1 .

**WARNING**

Make sure the printer is a serial printer before you continue, otherwise, you could damage the controller and the printer.

- Step**
- 1** Turn on the printer if you have not already done so.
 - 2** Set the default device to serial printer:
 - a** Press MENU.
 - b** Select FILE.
 - c** Press F1, [TYPE].
 - d** Select File.
 - e** Press F5, [UTIL].
 - f** Select Set Device.
 - g** Move the cursor to Serial Printer and press ENTER.
 - 3** Display the screen you want to print.
 - 4** Press FCTN key.
 - 5** Select PRINT SCREEN. The file will begin printing on the serial printer.

NOTE To pause printing, press PREV key.

9.3 MANIPULATING FILES

A file is a unit in which the system stores information. Files can be stored on a device attached to the controller port.

You perform file manipulations using the FILE menu. See the following screen for an example.

```

FILE JOINT 10%
FLPY:\*.* 1/16
1 * * (all files)
 2 * KL (all KAREL source)
 3 * CF (all command files)
 4 * TX (all text files)
 5 * LS (all KAREL listings)
 6 * DT (all KAREL data files)
 7 * PC (all KAREL p-code)
 8 * TP (all TP programs)
 9 * MN (all MN programs)
10 * VR (all variable files)
Press DIR to generate directory

11 * SV (all system files)
12 * IO (I/O config data)
13 * DF (all DEFAULT files)
14 * ML (all part model files)
15 * BMP (all bit-map images)
16 [you enter]
Press DIR to generate directory
[TYPE] [DIR] LOAD [BACKUP] [UTIL] >

DELETE COPY DISPLAY >

```

From the FILE menu you can:

- Generate a directory of files
- Load files from disk onto controller memory
- Back up program and system files
- Display text (ASCII) files
- Copy files to a disk
- Delete files from a disk
- Check and purge file memory
- Create error log files

Types of Files

To manipulate a file you must know the type of file you are manipulating. Table 9-9 lists several types of files available. During your work on the controller, you might only work with a few types of files. You can determine the file type by looking at the file name as it is displayed on the FILE menu. The file name consists of a file name, followed by a period, followed by a two-letter file type:

file.XX

where **file** is the file name and **XX** is the file type.

NOTE File types with three characters might be displayed on the FILE screen. These types are for various kinds of compressed files. The display of these file types is controlled by the system variable \$FILE_MASK. Refer to the *SYSTEM R-J2 Controller Software Reference Manual* for more information.

Table 9-9. Types of Files

File Type	Description
Bit map files (.BMP)	Bit map files contain bit map images used in robot vision systems.
Command file (.CF)	Command files are text (ASCII) files that contain a sequence of KCL commands for a command procedure.
Default file (.DF)	Default files are binary files that contain the default motion instructions for teach pendant programming.
Data file (.DT)	Data files are text (ASCII) or binary files that contain any data that is needed by the user.
I/O file (.IO)	I/O files are binary files that store configuration data.
KAREL file (.KL)	KAREL files are text (ASCII) files that contain the KAREL language statements for a KAREL program.
Listing file (.LS)	Listing files are text (ASCII) files that contain the listing of a KAREL language program, and line numbers for each KAREL statement. Listing files are also generated when a teach pendant screen is printed.
Part model files (.ML)	Part model files contain part model information used in robot vision systems.
P-Code file (.PC)	P-code files are binary files that contain the translated version of a .KL KAREL program file. This is the file that is actually loaded into controller memory and executed.
System file (.SV)	System files are binary files that store default values for system variable, servo parameter data, and mastering data.
Teach pendant program file (.TP)	Teach pendant program files are binary files that contain teach pendant instructions for teach pendant programs.
Text file (.TX)	Text files are text (ASCII) files that contain system-defined or user-defined text.
Variable file (.VR)	Variable files are binary files that contain variable data for a KAREL program.

9.3.1 Generating a Directory of Files

A directory is a list of files on a specific storage device. You can display a directory of files on the device connected to the port.

Directory Subsets

Some devices contain hundreds of files. You can display a directory of all files, or a subset of the files. When you generate a directory of files, you can choose from among the following file subsets:

- *.* – all files
- *.BMP – bit map image files
- *.KL – KAREL program files
- *.CF – command files
- *.TX – text files
- *.LS – listing files
- *.DT – data files
- *.ML – part model files
- *.PC – p-code files
- *.TP – teach pendant program files
- *.VR – variable files
- *.SV – system files
- *.IO – I/O files
- *.DF – default files
- **ASCII Files** – Text files, including files of type .KL, .CF, .TX, .LS, .DT, and .ML
- **Loadable Files** – Files that can be loaded into controller memory, including files of type .PC, .TP, .MN, .VR, .SV, .IO and .DF.

Use Procedure 9-15 to generate a directory of files.

Procedure 9-15 Generating a Directory of Files

- Step**
- 1 Press MENUS.
 - 2 Select FILE. You will see a screen similar to the following.

```

FILE JOINT 10%
FLPY:\*.* 1/16
1 * * (all files)
 2 * KL (all KAREL source)
 3 * CF (all command files)
 4 * TX (all text files)
 5 * LS (all KAREL listings)
 6 * DT (all KAREL data files)
 7 * PC (all KAREL p-code)
 8 * TP (all TP programs)
 9 * MN (all MN programs)
10 * VR (all variable files)
Press DIR to generate directory

11 * SV (all system files)
12 * IO (I/O config data)
13 * DF (all DEFAULT files)
14 * ML (all part model files)
15 * BMP (all bit-map images)
16 [you enter]
Press DIR to generate directory
[TYPE] [DIR] LOAD [BACKUP] [UTIL] >

DELETE COPY DISPLAY >

```

Currently accessing device

- 3 Press F2, [DIR]. You will see a screen similar to the following.

```

Directory Subset JOINT 10%
1 *.* 5 *.LS
 2 *.KL 6 *.DT
 3 *.CF 7 *.PC
 4 *.TX 8 -- next page --
FILE
1 * * (all files)
 2 * KL (all KAREL source)
 3 * CF (all command files)
 4 * TX (all text files)
 5 * LS (all KAREL listings)
 6 * DT (all KAREL data files)
Press DIR to generate directory
[ TYPE ] [ DIR ] LOAD [BACKUP] [UTIL ]

```

- 4 Select the subset of files you want to display and press ENTER. If you select *.SV to display all system files, see the following screen for an example.

FILE		JOINT	10%
FLPY:*.SV			1/12
1	SYSVARS	SV	(system file)
2	SYSSERVO	SV	(system file)
3	SYSMAST	SV	(system file)
4	SYSMACRO	SV	(system file)
5	SYSSPOT	SV	(system file)
6	*	KL	(all KAREL source)
7	*	CF	(all command files)
8	*	TX	(all text files)
9	*	LS	(all KAREL listing)
0	*	DT	(all KAREL data files)
[TYPE]	[DIR]	LOAD	[BACKUP] [UTIL] >

To select another subset of files, press F2, [DIR], and repeat Step 4.

9.3.2 Loading and Restoring Files from Disk To Controller Memory

Loading files allows you to load a file and all of its relevant data from disk into controller memory. You can load files into controller memory from a floppy disk.

Typically, you load a file from a disk when

- You want to modify a program (teach pendant program file, .TP) that is not currently in controller memory
- You want to execute a file (teach pendant program file, .TP, or KAREL p-code file, .PC)
- You want to load variable information the system needs to function (system file, .SV)
- You want to load variable information required for a KAREL program (.VR file)
- You want to load saved I/O configuration (.IO file)
- You want to load saved default motion instructions (.DF file)

Loadable Files

Loadable files are those files that can be loaded into controller memory. They are

- Teach pendant program files (.TP)
- KAREL p-code files (.PC)
- System files (.SV)
- Variable files (.VR)
- I/O configuration files (.IO)
- Default motion instruction files (.DF)

NOTE System files can be loaded only at controlled start. Refer to Appendix C, “BootROM Operations,” for more information.

Only these types of files can be loaded into controller memory. You can load a single file or a group of files. Use Procedure 9-16 to load files using the FILE menu.

Restoring Files

You **restore** files from a disk when you have previously backed up the files using BACKUP on the FILE screen (Section 9.3.3). You can restore the following groups of files if you have previously backed them up using BACKUP:

- System files
- Teach pendant programs
- Application files

Use Procedure 9-17 to restore BACKUP files using the FILE menu. This procedure will restore all files on the default device that were backed up using the BACKUP command. Refer to Section 9.3.3 for more information on backing up files.

Procedure 9-16 Loading Files Using the FILE Menu

⚠ CAUTION

Before you connect the floppy disk drive to the controller, turn on the controller, then connect and turn on the floppy disk drive; otherwise, equipment could be damaged.

- Condition**
- **If you are loading files from floppy disk**, the serial disk drive is connected to the controller port, is turned on, and contains the appropriate floppy disk.
 - The default device is set correctly. Refer to Procedure 9-4 .
- Step**
- 1 Press MENUS.
 - 2 Select FILE.
 - 3 Press F1, [TYPE].
 - 4 Select File.
 - 5 **To load a single file:**
 - a Generate a directory of the disk that contains the file you want to load. Refer to Procedure 9-15 .
 - b Move the cursor to the name of the file you want to load and press F3, LOAD. You will see a screen similar to the following.

FILE		JOINT	10%
FLPY:*. *		1/12	
1	PROG_1	TP	(TP program)
2	PROG_2	TP	(TPE program)
3	PROG_3	TP	(TPE program)
4	*	*	(all files)
5	*	KL	(all KAREL source)
6	*	CF	(all command files)
7	*	TX	(all text files)
8	*	LS	(all KAREL listing)
9	*	DT	(all KAREL data files)
Load FLPY:\PROG1		YES	NO

⚠ CAUTION

When you load or restore the file FRAMEVAR.SV, SYSVARS.SV, or SYSMAST.SV, make sure the motion configuration (items such as the number of motion groups and extended axes) of your system is the same as the motion configuration of the system on which the FRAMEVAR.SV, SYSVARS.SV, or SYSMAST.SV files were created. Otherwise, you might not be able to load or restore this file on an improperly configured system.

Loading, please wait

Loaded PROG_1.TP

PROG_1.TP already exists

OVERWRITE	SKIP	CANCEL
F3	F4	F5

6 Load the file(s):

- **To load the file(s) you selected**, press F4, YES.
- **If you do not want to load the file(s) you selected**, press F5, NO.

7 If the program already exists:

- **To overwrite**, press F3, OVERWRITE.
- **To skip** the file, press F4, SKIP.
- **To cancel**, press F5, CANCEL.

Procedure 9-17 Restoring BACKUP Files Using the FILE Menu

 **CAUTION**

Before you connect the floppy disk drive to the controller, turn on the controller, then connect and turn on the floppy disk drive; otherwise, equipment could be damaged.

Condition

- **If you are restoring files from floppy disk**, the serial disk drive is connected to the controller P2 port, is turned on, and contains the appropriate floppy disk.
- The device from which you want to restore files is set as the default device. Refer to Procedure 9-4 .

Step

- 1 Perform a controlled start as follows:
 - a **If the controller is turned on**, turn it off.
 - b On the teach pendant, press and hold the PREV and NEXT keys.
 - c While still pressing PREV and NEXT on the teach pendant, press the ON button on the operator box or operator panel.
 - d After the BMON> prompt appears on the teach pendant screen, release the PREV and NEXT keys.
 - e Press F2, CTRL, and press ENTER.
 - f Press F5, START, and press ENTER. This begins the controlled start. You will see a screen similar to the following.

BMON>

BMON> CTRL

BMON> START

Controlled Start Initialization

```

1 MOTION SYSVAR SETUP
2 PROGRAM INIT
3 MOTION DEVELOPMENT
4 EXIT

```

- 2 Press 4, EXIT.
- 3 Press F4, YES.
- 4 Press MENUS.
- 5 Select File.
- 6 Press F4, RESTORE.

Exit? [NO]

System Files 7 To restore system files, select System files. You will see a screen similar to the following.

- 1 **System files**
- 2 TP programs
- 3 Application
- 4 Applic.-TP
- 5 All of above

Restore from PS-100/200 Disk(OVRWRT)?

YES NO

⚠ CAUTION

In the next step, backed up files will be loaded and will overwrite existing files of the same name. Be sure you want to overwrite existing files before you restore them; otherwise, you could lose important data.

⚠ CAUTION

When you load or restore the file FRAMEVAR.SV, SYSVARS.SV, or SYSMAST.SV, make sure the motion configuration (items such as the number of motion groups and extended axes) of your system is the same as the motion configuration of the system on which the FRAMEVAR.SV, SYSVARS.SV, or SYSMAST.SV files were created. Otherwise, you might not be able to load or restore this file on an improperly configured system.

- a Restore the files:
 - To continue the restore, press F4, YES.
 - To cancel the restore, press F5, NO.

NOTE To cancel the restore at any time, press the PREV key.

The system will load all of the files that are listed in the \$FILE_SYSBCK system variable. You will not have to convert any variable files that have been restored. Any necessary conversion will be performed automatically.

- b If an error occurs during the restore, the restore will pause.
 - To skip the current file and continue restoring the remaining files, press F4, SKIP.
 - To cancel the restore from this file on, press F5, CANCEL.
- c When the restore has completed, you will see a message reporting the number of files restored. See the following screen for an example.

Total 9/9 files restored

Teach Pendant Program Files (.TP, .DF, .MN)

- 1 System files
- 2 **TP programs**
- 3 Application
- 4 Applic.-TP
- 5 All of above

8 To restore .TP, .DF, and .MN files, select TP programs. You will see a screen similar to the following.

Restore from PS-100/200 Disk(OVRWRT)?
YES NO

⚠ CAUTION

In the next step, backed up files will be loaded and will overwrite existing files of the same name. Be sure you want to overwrite existing files before you restore them; otherwise, you could lose important data.

- a Restore the files:
 - To continue the restore, press F4, YES.
 - To cancel the restore, press F5, NO.
- b If you answer YES, you will see a screen similar to the following.

OK to go to Control Start 2?
YES NO

- To continue the restore, press F4, YES.
- To cancel the restore, press F5, NO.

NOTE To cancel the restore at any time, press the PREV key.

- c If you answer YES, you will see a screen similar to the following.

30 seconds system save in progress..

The system will load all .TP, .DF, and .MN files. During the load you will see a screen similar to the following.

Loading DF_LOGI.DF (6 / 9)

- d If an error occurs during the restore, the restore will pause.
 - To skip the current file and continue restoring the remaining files, press F4, SKIP.
 - To cancel the restore from this file on, press F5, CANCEL.
- e When the restore has completed, you will see a message reporting the number of files restored. See the following screen for an example.

Total 8/9 files restored

- The label on the F4 function key will become [BACKUP].
- The controller will be in CONTROL START 2 startup mode.

Application Files 9 To restore non-program application files, select Application files. You will see a screen similar to the following.

```
1 System files
2 TP programs
3 Application
4 Applic.-TP
5 All of above
```

```
Restore from PS-100/200 Disk(OVRWRT)?
                                     YES    NO
```

⚠ CAUTION

In the next step, backed up files will be loaded and will overwrite existing files of the same name. Be sure you want to overwrite existing files before you restore them; otherwise, you could lose important data.

a Restore the files:

- To continue the restore, press F4, YES.
- To cancel the restore, press F5, NO.

NOTE To cancel the restore at any time, press the PREV key.

The system will load all of the files that are listed in the \$FILE_APPBCK system variable. You will not have to convert any variable files that have been restored. Any necessary conversion will be performed automatically.

b If an error occurs during the restore, the restore will pause.

- To skip the current file and continue restoring the remaining files, press F4, SKIP.
- To cancel the restore from this file on, press F5, CANCEL.

c When the restore has completed, you will see a message reporting the number of files restored. See the following screen for an example.

```
Total 3/3 files restored
```

Application Teach Pendant 10 Program Files (.TP, .DF, .MN)

- 1 System files
- 2 TP programs
- 3 Application
- 4 **Applic.-TP**
- 5 All of above

To restore application teach pendant programs (.TP, .DF, .MN) files, select Applic.-TP. You will see a screen similar to the following.

```
Restore from PS-100/200 Disk(OVRWRT)?
                                     YES    NO
```

CAUTION

In the next step, backed up files will be loaded and will overwrite existing files of the same name. Be sure you want to overwrite existing files before you restore them; otherwise, you could lose important data.

- a Restore the application teach pendant files:
 - To continue the restore, press F4, YES.
 - To cancel the restore, press F5, NO.
- b If you answer YES, you will see a screen similar to the following.

```
OK to go to Control Start 2?
                                     YES    NO
```

- To continue the restore, press F4, YES.
- To cancel the restore, press F5, NO.

NOTE To cancel the restore at any time, press the PREV key.

- c If you answer YES, you will see a screen similar to the following.

```
30 seconds system save in progress..
```

The system will load all files listed in the \$FILE_AP2BCK system variable. These files must be .TP, .DF., or .MN files.

- d If an error occurs during the restore, the restore will pause.
 - To skip the current file and continue restoring the remaining files, press F4, SKIP.
 - To cancel the restore from this file on, press F5, CANCEL.
- e When the restore has completed, you will see a message reporting the number of files restored. See the following screen for an example.

```
Total 1/2 files restored
```

- The label on the F4 function key will become [BACKUP].
- The controller will be in CONTROL START 2 startup mode.

- 1 System files
- 2 TP programs
- 3 Application
- 4 Applic.-TP
- 5 **All of above**

All Files 11 To restore all files, select All of above. You will see a screen similar to the following.

```
Restore from PS-100/200 Disk(OVRWRT)?
                                     YES    NO
```

⚠ CAUTION

In the next step, backed up files will be loaded and will overwrite existing files of the same name. Be sure you want to overwrite existing files before you restore them; otherwise, you could lose important data.

a Restore the files:

- To continue the restore, press F4, YES.
- To cancel the restore, press F5, NO.

NOTE To cancel the restore at any time, press the PREV key.

The system will load all of the files that are listed in the \$FILE_SYSBCK, \$FILE_APPBCK, and \$FILE_AP2BCK system variables, and all .TP, .DF, and .MN files. You will not have to convert any variable files that have been restored. Any necessary conversion will be performed automatically. During the restore, you will see a screen similar to the following.

```
Loading SYSVARS.SV    ( 4 / 21 )
```

b If an error occurs during the restore, the restore will pause.

- To skip the current file and continue restoring the remaining files, press F4, SKIP.
- To cancel the restore from this file on, press F5, CANCEL.

c When the restore has completed, you will see a message reporting the number of files restored. See the following screen for an example.

```
Total 20/21 files restored
```

- The label on the F4 function key will become [BACKUP].
- The controller will be in CONTROL START 2 startup mode.

12 To operate the robot, perform a cold start:

- a** Press FCTN.
- b** Select START (COLD).

9.3.3 Backing Up Program and System Files

You can *back up* program files, system files, application, and error log files to floppy disk, Flash ROM disk, or memory card using the FILE screen. When you back up a file you save it from controller memory to a disk, such as a floppy disk, so that you have a second copy of the file.

NOTE To back up all memory on the controller, use the Controller Backup and Restore function. Refer to Section 9.4.

Program Files

When you back up program files, all teach pendant program files currently loaded onto controller memory (listed on the SELECT menu) will be saved to the default device.

System Files

System files are binary files that store default values for system variable, servo parameter data, and mastering data. They contain information specific to the controller, robot, and software. When you backup system files, all system variable, servo parameter, and mastering data currently on controller memory is saved to the default device. The following are system files:

- **DIOCFGV.IO** – contains I/O configuration information
- **FRAMEVAR.SV** - contains frame information
- **NUMREG.VR** - contains register information
- **POSREG.VR** - contains position register information
- **SYSVARS.SV** – contains system variable default values for your system.
- **SYSSERVO.SV** – contains servo parameter data the robot needs to function. The values in this file are loaded automatically when the controller is turned on.
- **SYSMAST.SV** – contains dynamic mastering data, which is automatically created when the robot is mastered.
- **SYSPASS.SV** – contains password setup information.
- **SYSMACRO.SV** – contains dynamic mastering data, which is automatically created when the robot is mastered.

When you back up system files, all five files are copied to the default device.

Application Files

Application files are all program variable files.

Application Teach Pendant Program Files

Application teach pendant program files are teach pendant program files with file type .TP, .DF, or .MN. The names of the application TP files are stored in the system variable \$FILE_AP2BCK.

Error Log Files

Error log files are ASCII files that give a snapshot of the errors in the system. They can be backed up to the default device, but cannot be restored or loaded into the controller. Two kinds of error log files are backed up: ERRALL.LS and ERRACT.LS. Refer Table 9-10 to for descriptions of these files.

Table 9-10. Error Log Files

File	Description	Menu
ERRALL.LS	Contains a snapshot of the history of errors in the system	ALARM, [TYPE], Alarm Log, Hist menu
ERRACT.LS	Contains a snapshot of the active errors in the system	ALARM, [TYPE], Alarm Log, Active menu

The information in an error log file follows a specific format, which is shown as follows. The first line is the error log *header*, and subsequent lines are *error entries*.

Header:

```
S1:\ERR_ALL.LS Robot Name PALROB Time: 17:21:26
08/28/97
```

The **header** consists of the error log file name, robot host name, the name of the currently selected program or file, and the current system time and date.

Error Entry:

```
255" 10-SEP-97 10:35 " SRVO-154 HVAL(CNV-DC) alarm
(G:1 A:4)" " SERVO" act "
```

Each **error entry** consists of the following:

- Sequence number – internal system number that identifies a particular error in the error log
- Date and time
- Error facility name
- Error code number
- Error code message
- Cause code message, if applicable
- Severity text
- Active/inactive status of the alarm, for ERRALL.LS only – indicates whether the alarm is currently active. “act” indicates that the alarm is currently active. No text indicates that the alarm is not active.

Use Procedure 9-18 to back up program and system files to disk.

Procedure 9-18 Backing Up System Files, TP Programs, and Application Files to Disk

CAUTION

Before you connect the floppy disk drive to the controller, turn on the controller, then connect and turn on the floppy disk drive; otherwise, equipment could be damaged.

- Condition**
- **If you are backing up files to a floppy disk**, the serial disk drive is connected to the controller port, is turned on, and contains a formatted disk.
 - The default device is set correctly. Refer to Procedure 9-4 .
- Step**
- 1 Press **MENUS**.
 - 2 Select **FILE**.
 - 3 Press **F1**, [**TYPE**].
 - 4 Select **File**.
 - 5 Press **F4**, [**BACKUP**].

- System Files**
- 1 **To back up only system files**, select System files. You will see a screen similar to the following.

```

1 System files
2 TP programs
3 Application
4 Applic.-TP
5 Error log
6 All of above
7 Controller*

```

* for ECBR-FTP option only

```

Save FLPY:\DIOCFGSV.IO?
EXIT      ALL      YES      NO

```

The first system file name in the system variable \$FILE_SYSBCK will be displayed.

- 2 Back up the specified files:
 - **To back up this system file only**, press **F4**, **YES**.
 - **If you do not want to back up this system file**, press **F5**, **NO**. The next system file name in controller memory will be displayed.
 - **To back up all system files**, press **F3**, **ALL**. If the file already exists, then you will have the option to overwrite, skip, or cancel.
 - **To exit**, press **F2**, **EXIT**.

Teach Pendant (TP) Programs

```
1 System files
2 TP programs
3 Application
4 Applic.-TP
5 Error log
6 All of above
7 Controller*
```

* for ECBR-FTP option only

1 To back up only teach pendant program files, select TP Programs. The first program name in controller memory (the SELECT menu) will be displayed.

- **To back up this program only**, press F4, YES.
- **If you do not want to back up this program**, press F5, NO. The next program name in controller memory will be displayed.
- **To back up all teach pendant programs**, press F3, ALL. If the file already exists, then you will have the option to overwrite, skip, or cancel.
- **To exit**, press F2, EXIT.

Application Files

```
1 System files
2 TP programs
3 Application
4 Applic.-TP
5 Error log
6 All of above
7 Controller*
```

* for ECBR-FTP option only

1 To back up only application files other than Teach Pendant program files, select Application.

The first application file name in the \$FILE_APPBCK system variable will be displayed.

- **To back up this application file only**, press F4, YES.
- **If you do not want to back up this application file**, press F5, NO. The next application file name in controller memory will be displayed.
- **To back up all application files**, press F3, ALL. If the file already exists, then you will have the option to overwrite, skip or cancel.
- **To exit**, press F2, EXIT.

Application Teach Pendant Program Files (.TP, .DF, .MN)

```
1 System files
2 TP programs
3 Application
4 Applic.-TP
5 Error log
6 All of above
7 Controller*
```

* for ECBR-FTP option only

1 To back up only application teach pendant program files (.TP, .DF, .MN), select Applic.-TP.

The first application teach pendant program name in the \$FILE_AP2BCK system variable will be displayed.

- **To back up this program only**, press F4, YES.
- **If you do not want to back up this program**, press F5, NO. The next program name in controller memory will be displayed.
- **To back up all application teach pendant programs**, press F3, ALL. If the file already exists, then you will have the option to overwrite, skip or cancel.
- **To exit**, press F2, EXIT.

Error Log Files

```
1 System files
2 TP programs
3 Application
4 Applic.-TP
5 Error log
6 All of above
7 Controller*
```

* for ECBR-FTP option only

1 To back up only error log files, select Error log.

The first error log file name in the system variable \$FILE_ERRBCK will be displayed.

- **To back up this error log file only**, press F4, YES.
- **If you do not want to back up this error log file** press F5, NO. The next error log file name from the system variable \$FILE_ERRBCK will be displayed.
- **To back up all error log files (from \$FILE_ERRBCK)**, press F3, ALL. If the file already exists, then you will have the option to overwrite, skip, or cancel.
- **To exit**, press F2, EXIT.

All Files

```

1 System files
2 TP programs
3 Application
4 Applic.-TP
5 Error log
6 All of above
7 Controller*

```

* for ECBR-FTP option only

- 1 To back up all types of files**, select All of above. You will see the following message displayed at the bottom of the screen.

```

Del PS-100/200 Disk, backup all files?
                                YES    NO

```

NOTE All files on floppy disk or memory card are deleted before this type of backup. If the destination device is networked, files will not be deleted. The file number and total number of files are displayed during backup.

```

Backing up SYSVARS.SV (8/21)

```

- **If you do not want to delete the files on the default device and then back up the files**, press F5, NO. The files will not be backed up.
- **If you want to delete the files on the default device and back up the specified files** press F4, YES. All files in the \$FILE_SYSBCK, \$FILE_APPBCK, and \$FILE_ERRBCK system variables will be backed up. All .TP and .DF files will also be backed up.

NOTE If an error occurs while the files are being saved, you will be prompted with a message and asked if you want to proceed.

When the backup is complete, the FILE menu will be displayed and you can display a directory of the default device by pressing DIR. In addition, a date and time file called BACKDATE.DT will be created on the default device. This file contains the date and time the backup of all files was performed.

NOTE For information on backing up controller files, refer to the *Ethernet Controller Backup and Restore – FTP Setup and Operations Manual*.

- Restore Backup Files** **1 To restore backup files**, you must load them. Refer to Procedure 9-16 or Procedure 9-17 .

9.3.4 Displaying Text (ASCII) Files

Displaying a text file shows the contents of an ASCII file on the screen. You can display the contents of only ASCII files. Loadable files cannot be displayed.

Displayable (ASCII) Files

Displayable files are ASCII or text files. They are

- KAREL program files (.KL)
- Command files (.CF)
- Text files (.TX)
- Listing files (.LS)
- Data files (.DT)
- Part model files (.ML)

Use Procedure 9-19 to display a text (ASCII) file.

Procedure 9-19 Displaying the Contents of a Text (ASCII) File

- | | |
|------------------|--|
| Condition | <ul style="list-style-type: none"> ■ If you are displaying the contents of a file from floppy disk, the serial disk drive is connected to the controller port, is turned on, and contains the appropriate disk. |
| Step | <ol style="list-style-type: none"> 1 Set the default device: <ol style="list-style-type: none"> a Press MENUS. b Select FILE. c Press F1, [TYPE]. d Select File. e Press F5, [UTIL]. f Select Set Device. g Move the cursor to the device you want and press ENTER. 2 Press MENUS. 3 Select FILE. 4 Press F1, [TYPE]. 5 Select File. 6 Generate a directory that displays the name of the file you want to display. 7 Move the cursor to the name of the ASCII or text file you want to display. 8 Press NEXT, > and press F3, DISPLAY. The file will be displayed on the screen. 9 To continue displaying, press F4, YES, otherwise press F5, NO. 10 When the file is finished being displayed, press any key to continue. |

Continue displaying?	YES	NO
----------------------	-----	----

Press any key to exit

9.3.5 Copying Files to a Disk

You can copy a single file or a subset of files from one file device to another. These devices include floppy disk, Flash ROM disk, and memory card. Use Procedure 9-20 to copy files to a disk.

Procedure 9-20 Copying Files to a Disk

 **CAUTION**

Before you connect the floppy disk drive to the controller, turn on the controller, then connect and turn on the floppy disk drive; otherwise, equipment could be damaged.

- Condition**
- **If you are copying files to a floppy disk**, the serial disk drive is connected to the controller P2 port, is turned on, and contains a formatted disk.
 - **If you are copying files to a memory card**, the memory card is installed properly. Refer to Section 9.1.3 for more information.
- Step**
- 1 Set the default device:
 - a Press **MENUS**.
 - b Select **FILE**.
 - c Press **F1**, [**TYPE**].
 - d Select **File**.
 - e Press **F5**, [**UTIL**].
 - f Select **Set Device**.
 - g Move the cursor to the device you want and press **ENTER**.
 - 2 Press **MENUS**.
 - 3 Select **FILE**.
 - 4 Press **F1**, [**TYPE**].
 - 5 Select **File**.

- 6 To copy a group of files**, move the cursor to the subset of files you want to copy and press NEXT, >, and then press F2, COPY.

To copy a single file, generate a directory that displays the file name, move the cursor to the name of the file you want to load, and press NEXT, >, and then press F2, COPY.

You will see a screen similar to the following.

FILE Copy		JOINT 10%
FLPY:\		
From:	FLPY:\SYSVARS.SV	
To Device:	***	
To Directory:	\	
To Filename:	SYSVARS.SV	
DO_COPY [CHOICE] CANCEL		

- 7** Press F4, [CHOICE], to select the device to which the file will be copied. You will see a screen similar to the following.

To Device	JOINT 10%
1 Floppy Disk	
2 FROM Disk (FR:)	
3 FTP (C1:)	
4 Memory Card (MC:)	
FILE Copy	
From:	FLPY:\SYSVARS.SV
To Device:	***
To Directory:	\
To Filename:	SYSVARS.SV [CHOICE]

- 8** Move the cursor to the device name you want and press ENTER. You will see a screen similar to the following.

FILE Copy		JOINT 10%
FLPY:\		
From:	FLPY:\SYSVARS.SV	
To Device:	FR:\	
To Directory:	\	
To Filename:	SYSVARS.SV	
DO_COPY CHANGE CANCEL		

- 9 To change the name of the filename to which the selected file will be copied, press F4, CHANGE. You will see a screen similar to the following.

```

1 Words
2 Upper Case
3 Lower Case
4 Options          --Insert--
Select
FLPY:\
From:             FLPY:\SYSVARS.SV
To Device:        FR:
To Directory:     \
To Filename:      SYSSERVO.SP

JOB          PROC          MOV_          TEST

```

- 10 Type the new file name and press ENTER.
- 11 Change information if necessary:
- **If you want to change any information on the screen**, select the desired field and enter the new information.
 - **If all the information on the screen is correct**, press F1, DO_COPY.
- 12 If file exists,
- To overwrite, press F4, YES, otherwise press F5, NO. When the copy is complete, the FILE menu is displayed.

```

Overwrite?
                YES  NO
Copying, please wait...

```

9.3.6 Deleting Files from a Disk

Deleting a file means permanently removing the file from a floppy disk. Once you delete a file, you cannot recover it.

Deleting a file from the FILE menu deletes the file from the default device. However, it does not delete it from controller memory. To delete a file from controller memory (the SELECT menu), refer to Procedure 9-12 . Use Procedure 9-21 to delete files from a disk.

Procedure 9-21 Deleting Files from a Disk

Condition ■ **If you are deleting files from a floppy disk**, the serial disk drive is connected to the controller port, is turned on, and contains a formatted disk.

- Step**
- 1 Set the default device:
 - a Press MENU.
 - b Select FILE.
 - c Press F1, [TYPE].
 - d Select File.
 - e Press F5, [UTIL].
 - f Select Set Device.
 - g Move the cursor to the device you want and press ENTER.
 - 2 Press MENU.
 - 3 Select FILE.
 - 4 Press F1, [TYPE].
 - 5 Select File.

 **CAUTION**

Make sure the default device is set to the device from which you want to delete the file(s); otherwise, you could delete the wrong files.

- 6 **Generate a directory** of the device from which you want to delete the file. Refer to Procedure 9-15 .
- 7 **To delete a group of files**, move the cursor to the subset of files you want to delete and press NEXT, >, and then press F1, DELETE. **To delete a single file**, move the cursor to the name of the file you want to delete, and press NEXT, >, and then press F1, DELETE.
- 8 Delete the file(s):
 - **To delete the specified file(s)**, press F4, YES.
 - **If you do not want to delete the specified file(s)**, press F5, NO.

```
Delete FLPY:\*.TP?
                YES  NO
```

```
Delete FLPY:\PROG_1.TP?
                YES  NO
```

9.3.7 Saving Files

Saving files allows you to save variables and other data to the default device. The following information can be saved using the SAVE function:

- **System variables** will be saved to **sysvars.sv**. The SYSTEM: SYSTEM Variables screen must be displayed to save data to this file.
- **Mastering information** will be saved to **sysmast.sv**. The SYSTEM: Master/Cal screen must be displayed to save data to this file.

NOTE To save servo parameters and other system files, use the BACKUP function. Refer to Procedure 9-18 .

- **Macro setup** information will be saved to **sysmacro.sv**. The SETUP Macro screen must be displayed to save data to this file.
- **Frame setup** information will be saved to **framevars.sv**. One of the SETUP Frame screens must be displayed to save data to this file.
- **I/O configuration** information will be saved to **diocfgsv.io**. One of the I/O screens must be displayed to save data to this file.
- **Register** information will be saved to **numreg.vr**. The DATA Registers screen must be displayed to save data to this file.
- **Position register** information will be saved to **posreg.vr**. The DATA Position Reg screen must be displayed to save data to this file.
- **SERVO parameters** information will be saved to **syssservo.sv**. The DATA Position Reg screen must be displayed to save data to this file.
- **Password information** will be saved to **syspass.sv**. The SETUP Passwords screen must be displayed to save data to this file.

 **CAUTION**

If the file you are saving already exists on the default device, saving the data using the SAVE function will **not** update the file. If you want to save the new file, first delete it from the default device then try saving it again.

Use Procedure 9-22 to save files.

Procedure 9-22 Saving Files to the Default Device

Condition ■ If you are saving program data, the program you want is the default program.

Step 1 Display the screen that contains the information you want to save. Refer to Table 9-11.

Table 9-11. Valid SAVE Function Screens

To Save This Data	Display This Screen	Saved To
All System Variables	SYSTEM SYSTEM Variables	SYSVARS.SV
Mastering Data	SYSTEM Master/Cal	SYSMAST.SV
Macro setup information	SETUP Macro	SYSMACRO.SV
Frame setup comment and setup information NOTE: The frame transforms are saved only when system variables are saved.	SETUP Frame	FRAMEVAR.SV
Input/Output current port assignment, mode, and port comment information	I/O (any digital screen)	DIOCFGSV.IO
Register values	DATA Registers	NUMREG.VR
Position register values	DATA Position Reg	POSREG.VR
Servo parameter data	SYSTEM SYSTEM Variables	SYSSEVERO.SV
Password data	SETUP Passwords	SYSPASS.SV

For example, to save system variable information:

- a Press **MENU**.
- b Select **SYSTEM**.
- c Press **F1**, [**TYPE**].
- d Select **Variables**. You will see a screen similar to the following.

SYSTEM Variables		JOINT 50%
		1/125
1 \$ANGTOL	9 of REAL	
2 \$APPLICATION	[3] of STRING [21]	
3 \$AP_MAXAX	0	
4 \$AP_PLUGGED	2	
5 \$AP_TOTALAX	16777216	
6 \$AP_USENUM	[32]of BYTE	
7 \$ASCII_SAVE	FALSE	
8 \$AUTOINIT	2	
9 \$BLT	0	
10 \$CHECKCONFIG	FALSE	
[TYPE]		

- e Press the **FCTN** key to display the miscellaneous menu.
- f Select "0 —NEXT—" and "2 Save".

9.3.8 Checking and Purging File Memory

You can check the amount of memory you are using in the file system using the File Memory screen. In addition, you can purge unused memory space on the Flash ROM disk.

The purge operation is necessary only when the Flash ROM disk does not have enough memory to perform an operation, such as copy or save. When you perform a purge, the system will erase file blocks that were previously used, but no longer needed. These are called *garbage blocks*. The Flash ROM disk might contain many garbage blocks if files are deleted or overwritten frequently.

When you perform a purge, the device must be mounted and no files on the Flash ROM disk can be open.

Procedure 9-23 Checking and Purging File Memory

- Step**
- 1 Press MENUS.
 - 2 Select FILE.
 - 3 Press F1, [TYPE].
 - 4 Select File Memory. You will see a screen similar to the following.

FILE Memory		JOINT	50%
Device	Total	Free	
RD:	64.0 KB	64.0 KB	
FR:	960.0 KB	261.0 KB	
[TYPE]		PURGE	HELP

NOTE Before you perform a purge, make sure that no files are open on the Flash ROM disk. Otherwise, an error will occur.

- 5 **To purge unused memory**, press F4, PURGE. See the following screen for an example.

FILE Memory		JOINT	50%
Device	Total	Free	
RD:	64.0 KB	64.0 KB	
FR:	888.0 KB	359.0 KB	
Recoverable FR Kbytes:		2.0 KB	
Purge memory file device?		YES	NO

- 6 Select whether to purge the device:
 - To purge the device, press F4, YES.
 - To cancel the purge, press F5, NO.

9.4 CONTROLLER BACKUP AND RESTORE

Controller backup and restore allows an R-J2 controller to back up and restore controller memory. This capability is divided into two parts:

- Controller backup
- Controller restore

Controller backup is performed at controlled 2 start. During controller backup, the entire contents of controller memory are copied to files on the designated device. Refer to Section 9.4.1.

Controller restore is performed from the Boot Monitor (BMON). During controller restore, all of FROM and CMOS is cleared and then files previously created using the controller backup procedure are loaded from the default device. Refer to Section 9.4.2.

NOTE FTP can be used to transfer controller memory files over an Ethernet network. The Trivial File Transfer Protocol (TFTP) can be used to load controller memory files onto the controller via an Ethernet network. Refer to the *SYSTEM R-J2 Ethernet Controller Backup Restore/FTP Setup and Operations Manual* for more information.

CAUTION

If you restore a controller and a file already exists on the controller, the file is automatically overwritten.

9.4.1 Backing up a Controller

The backup feature allows you to back up the entire contents of controller memory. The backup procedure sets up the files so that controller memory can be fully restored if necessary. When you restore the controller backup to the controller, you will have a fully loaded controller.

Use Procedure 9-24 to perform a controller backup using a floppy or a memory card device. To perform a controller backup using Ethernet, refer to the *SYSTEM R-J2 Ethernet Controller Backup/FTP Setup and Operations Manual*.

The controller backup procedure creates files with file extension *.ldc*.

Backup Files

When a controller backup is performed, the controller memory is copied into compressed binary image files with the file extension *.ldc*. The backup utility will create the number of files required for backup storage. By default, these files will be named *backup##.ldc*, where *##* is the file number. For example, if the backup creates two files, the files will be named *backup01.ldc* and *backup02.ldc*.

A minimum of two backup files will be created: one for controller FROM and one for controller CMOS.

During the backup process, you will have the option to change the name of the backup files from *backup* to the name you want. If you are doing backups of several controllers, you might want to name the files using the F number of the robot.

In addition to creating the backup files, the backup utility creates a .cf file for each memory card or floppy disk. The first one is called *restore.cf*. The rest of the files will have unique names based on the date and time stamp from when the backup was performed. When a controller restore is performed, these files are used to direct the system to load all of the files created during the backup.

 **CAUTION**

The *restore.cf* file is overwritten each time a controller backup is performed, regardless of the names of the controller backup files. If you are backing up more than one controller, create a separate subdirectory to contain backup files and *restore.cf* for each controller. Otherwise *restore.cf* will be overwritten and you will not be able to restore controller memory.

You might want to store the controller backup files in the location from which you will load them. It is a good idea to create a separate subdirectory for each robot. If you store backups

- On a **UNIX workstation**, the load directory usually is the **/usr** directory on the local hard drive of the workstation. This is due to restrictions on file access established by some TFTP server implementations.
- On a **personal computer**, the load directory can be any directory you specify.

Use Procedure 9-24 to perform a controller backup to a floppy or memory card device.

Procedure 9-24 Backing up a Controller to a Floppy or Memory Card Device

NOTE If an error occurs during controller backup, correct the error and try to continue. If the system does not allow you to continue, repeat the entire controller backup procedure.

- Step 1** Perform a controlled 2 start.
- a** If the controller is turned on, turn it off.
 - b** On the teach pendant, press and hold the PREV and NEXT keys and press the ON button.
 - c** After the BMON> prompt appears on the teach pendant screen, release the PREV and NEXT keys.
 - d** Press F2, CTRL, and press ENTER.
 - e** Press F5, START, and press ENTER. This begins the controlled start. You will see a screen similar to the following.

BMON>

BMON> CTRL

BMON> START

```

Controlled Start Initialization

1 MOTION SYSVAR SETUP
2 PROGRAM INIT
3 MOTION DEVELOPMENT
4 EXIT

Press enter or number key to select.
  
```

- f** Select 4, EXIT, and press ENTER.
- g** Press F4, YES.
- h** Press FCTN.

⚠ CAUTION

The CTRL2 start takes a thirty seconds to finish. Do not turn off the controller until the CTRL2 start has completed. Otherwise, you will lose the software loaded on your controller and will have to reload it. The CTRL2 start is finished when the FCTN menu disappears and you can display it again by pressing the FCTN key.

- i** Select START (CTRL2) and press ENTER. The CTRL2 start will be performed. When it is finished, you will see a title line on the screen similar to the following.

```

CONTROL START 2 MENUS
  
```

```

1 Floppy disk
2 Serial Printer
3 FROM Disk (FR:)
4 FTP (CI:)
5 Memory Card (MC:)

```

- 2 Press MENUS.
- 3 Select File.
- 4 Press F5, [UTIL].
- 5 Select Floppy disk or Mem Card (MC:).
- 6 Press F4, [BACKUP].

CAUTION

Backing up files to a memory card or floppy disk will erase all the information on the card or disk before the backup is performed. You will lose all the information currently stored on the memory card or floppy disk.

```

1 System files
2 TP programs
3 Application
4 All of above
5 Controller

```

- 7 Select Controller. You will see a screen similar to the following.

If you have selected the **Memory Card Device** you will see this screen

```

FILE Backup          CONTROL START 2 MENUS

Controller backup will backup the
controller's memory to compressed load
files on memory cards. Insert a
memory card.

WARNING: Any files on the card will
be lost.

Press CONTINUE when ready.
                        CONTINUE  CANCEL

```

If you have selected the **Floppy Device** you will see this screen

```

FILE Backup          CONTROL START 2 MENUS

Controller backup will backup the
controller's memory to compressed load
files on disks. Insert a disk.

WARNING: Any files on the disk will
be lost.

Press CONTINUE when ready.
                        CONTINUE  CANCEL

```

To continue, press F4, CONTINUE. You will see a screen similar to the following.

If you have selected the **Memory Card Device** you will see this screen

```

FILE Backup          CONTROL START 2 MENUS

Backup may require  2 2MB memory cards
or 36666 KB of storage and will take
approximately 15 minutes per card.

WARNING: Any files on the card will be
lost.

Root name for .LDC files:  backup
PREV to Cancel

Press CONTINUE when ready.

                                CONTINUE  CANCEL

```

If you have selected the **Floppy Device** you will see this screen

```

FILE Backup          CONTROL START 2 MENUS

Backup may require  6 720KB disks
or 36666KB of storage and will take
approximately 20 minutes per disk.

WARNING: Any files on the disks
will be lost.

Root name for .LDC files:  backup
PREV to Cancel

Press CONTINUE when ready.

                                CONTINUE  CANCEL

```

8 To enter a root name other than the default (backup) for the compressed files that will be created, do the following:

a Press ENTER. You will see a screen similar to the following.

```

                                CONTROL START 2 MENUS

1 Upper Case
2 Lower Case
3 Punctuation
4 Options          --Insert--
FILE Backup

Root name for .LDC files:  backup
PREV to Cancel

Old Value:  backup
ABCDEF GHIJKL MNOPQR STUVWX YZ @*.

```

b Use the function keys to enter the root name and press ENTER.

- 9** To continue, press F4, CONTINUE.
If you do not want to continue the backup, press F5, CANCEL.

The system will start to write backup files.

If you want to cancel, press PREV. However, the system will not respond until it has completed writing the current file.

- 10** When the system has finished writing the current .ldc files, and a second memory card is required, you will see one of the following messages:

If you have selected the **Memory Device** you will see this screen

```
Insert a memory card for the LDC files
                                CONTINUE  CANCEL
```

If you have selected the **Floppy Device** you will see this screen

```
Insert a disk for the LDC files
                                CONTINUE  CANCEL
```

- 11** Insert the next memory card or floppy disk and press F4, CONTINUE.
- 12** When the system has finished writing the current .ldc files, you will see the following message:

```
Controller backup completed successfully
```

- 13** To exit the screen, press PREV.

9.4.2 Restoring a Controller

The controller restore function allows you to restore controller memory on a controller from a backed up controller. Use Procedure 9-25 to restore a controller.

CAUTION

If you restore a controller and a file already exists on the controller, the file is automatically overwritten.

Procedure 9-25 Restoring a Controller after a Backup

- Condition**
- You have the controller backup files on memory cards or floppy disks. (Procedure 9-24 must have been performed before you can restore a controller.)
 - Make sure the Ethernet card has not been started. Otherwise, the restore process will try to access the network.
 - If you are restoring from memory cards, make sure the floppy is disconnected.

CAUTION

You must use Procedure 9-24 to back up a controller before you can restore a controller using this procedure. Otherwise, the controller restore procedure will not function properly.

- Step**
- 1 The controller is turned off.
1 Press and hold the PREV and NEXT keys on the teach pendant, then press the ON button.

The boot monitor prompt, BMON>, is displayed. You will see a screen similar to the following.

```

*** BOOT MONITOR for R-J2 CONTROLLER ***
Version 4.22                               01-JAN-199x

F-ROM/D-RAM/C-MOS : 6.0/8.0/1 MB
TP Version         : I
Current TIME       : 01-JAN-199x 22:52:53

Slot  ID      FC      OP
0     9B      0       0     R-J2 Main CPU
1     6A      0       0     AB/Ether I/F optional
D     6A      0       0     MCARD I/F optional

BMON>
COLD   CTRL   INIT   NOLOAD  START  >

```

- 2 Insert the first memory card or floppy disk in the memory card interface or floppy disk drive.
- 3 Press NEXT, >, until F2, INSTALL, is displayed.
- 4 Press F2, INSTALL, and press ENTER.
- 5 Press NEXT, >, until F5, RESTORE, is displayed.

```
BMON> INSTALL
INSTALL>
LOAD    FSLOAD    RUN    ENET    RESTORE >
```

- 6 Press F5, RESTORE, and press ENTER.
You will be asked to run restore.cf.
- 7 **If you want to continue**, press 1 and then press ENTER.
To cancel, press 0 and then press ENTER.

Restoring will take several minutes (approximately 2–3 minutes per memory card).

```
Restore complete. Power off,
then on.
```

When the INSTALL prompt is displayed, a message will be displayed stating that power must be cycled for the restore to take effect.

- 8 Turn off the controller and then turn it on. The controller will start up in CONTROLLED START mode. A message will be displayed stating that the Cleanup of FROM blocks is occurring.

10 ADVANCED FUNCTIONS

Topics In This Chapter		Page
Mirror Image Utility	Mirror image allows you to flip the positions of an entire teach pendant program or a portion of a program, creating a mirror image of the original program or program portion.	10-4
Program Shift Utility	Program shift allows you to offset the positions of an entire teach pendant program or a portion of a teach pendant program.	10-16
Space Check Function	The space check function, when incorporated into a robot, monitors a predetermined interference area. When another robot or a peripheral unit is located within that area, the function stops the operation of the robot if a move command specifying movement into that area is issued to the robot.	10-26
Executing Multiple Programs (Multi-Tasking)	Multi-tasking allows you to run more than one program on the controller at the same time using a time-sharing basis. The maximum number of user programs that can be executed simultaneously is four; the default is one. <ul style="list-style-type: none"> • Guidelines • Synchronizing the Execution of Multiple Programs • Affect of Multi-tasking on Dedicated I/O Signals • Standard Operator Panel (SOP) Cycle Start Execution • Program Number Select (PNS) Execution • RUN Program Instruction Execution • Single Step Program Execution 	10-30 10-30 10-31 10-31 10-32 10-33 10-34 10-35
Angle Entry Shift Function	The angle entry shift function performs a programmed shift according to three or four representative taught points and directly entered rotation angles.	10-37
Reference Position Utility	Reference position allows you to specify a joint position and then assign a digital output or robot output signal to that position. When the robot moves to within a tolerance range of the joint position, the assigned digital output (DO) or robot output (RO) signal turns on.	10-43
Position Register Look-Ahead Execution Function	The position register look-ahead function enables look-ahead execution for position registers. <ul style="list-style-type: none"> • Program Instructions • Program Example • Execution 	10-46 10-47 10-48 10-49
Shape-Generation (option)	The shape generation option simplifies cutting two dimensional shapes. This shape generation software reduces the on-line programming time by reducing the number of required taught robot positions and providing features like shape cut macros and on-the-fly shape shift. <ul style="list-style-type: none"> • Shape Setup • Shape Schedules • Programming • Teach and Production Modes • Shape Adjust Utility • Shape Frames 	10-50 10-51 10-53 10-65 10-70 10-71 10-75
Motion Group DO Output Function	The motion group DO output function outputs information about motion groups that can be jogged or motion groups a program that is running or paused uses, as a digital output signal (DO) or a robot output signal (RO). <ul style="list-style-type: none"> • Restrictions • Operations • Setup • Subprograms and Multi-tasking Execution 	10-76 10-76 10-76 10-78

Topics In This Chapter		Page
Motion Start Delay Detection Function	<p>The motion start delay detection function detects and displays delay in a motion start. If deceleration or oscillation occurs, it is possible to determine the program line to be corrected according to the information detected by this function.</p> <ul style="list-style-type: none"> ● Restrictions 10-79 ● Detection Information and System Variables 10-80 ● Motion Start Delay Detection Function Example 10-82 	10-79
Soft Float Function (option)	<p>The soft float function is a feature that is used to compensate for variances in workpiece precision in applications where the robot is used to mount workpieces on a machine tool.</p> <ul style="list-style-type: none"> ● Soft Float Function Restrictions 10-84 ● Soft Float Schedules 10-86 ● Soft Float Program Instructions 10-90 	10-83
Continuous Turn Function	<p>The continuous turn function allows the last axis and extended rotation axis of the robot to turn in a given direction continuously and indefinitely.</p> <ul style="list-style-type: none"> ● Function 10-92 ● Setting 10-93 ● Motion Instructions 10-95 ● Operation 10-95 ● Example 10-96 ● Notes and Restrictions 10-97 ● Alarm Codes 10-98 	10-92
CRT Function	<p>The CRT function enables a factory terminal to be connected to the R-J2 controller. The factory terminal can provide almost the same display as a teach pendant and enables using a full keyboard.</p> <ul style="list-style-type: none"> ● Operation 10-102 ● Troubleshooting 10-105 	10-99
Singularity Check Function	<p>The singularity check function is provided to check whether the taught position is at a singular position.</p>	10-106
All-Point Teaching for Palletizing	<p>This function enables a robot to palletize or depalletize workpieces according to the configurations it has been taught, simply by setting a system variable.</p>	10-107
Coordinates Offset Function	<p>The coordinates offset function changes either the tool coordinate system (the tool itself) or the user coordinate system, for a range of motion instructions in a program for which teaching has been completed, then converts the position data such that the TCP position does not change, based on the shift between the original and changed coordinate system</p> <ul style="list-style-type: none"> ● Tool Frame Offset Function 10-114 ● User Frame Offset Function 10-117 	10-110
TIME BEFORE/AFTER Motion Option Instruction	<p>The time before/after motion option instruction allows you to specify a teach pendant program that is to be called at a specified time before or after the completion of a motion instruction.</p> <ul style="list-style-type: none"> ● Program Execution 10-121 ● Execution Timing 10-122 ● Recording a TIME BEFORE/AFTER Instruction 10-123 ● TIME BEFORE/AFTER Instruction Program Example 10-125 ● Programming Hints 10-126 	10-121

Topics In This Chapter		Page
Condition Monitor Function	The Condition Monitor Function monitors the condition of an I/O signal, register value, or alarm status, during program execution. As soon as the condition is triggered, the specified program is executed and interrupts the current program.	10-127
	<ul style="list-style-type: none"> ● Monitors ● Monitor State ● Monitor Instructions ● Condition Handler Program ● Conditions ● Condition Menu ● Restrictions 	10-128 10-129 10-130 10-130 10-131 10-132 10-134
Collision Guard (option)	The Collision Guard function provides a highly sensitive method to detect that the robot has collided with an object and stop the robot immediately. This helps to minimize the potential for damage to the end-of-arm tooling and robot.	10-141
	<ul style="list-style-type: none"> ● Limitation ● Falsely Detected Collisions ● Collision Guard Adjust Macro Program ● Setup ● Programmed Motion 	10-141 10-142 10-142 10-143 10-145
Error Recovery (option)	Error Recovery allows you to specify how the robot will recover from errors automatically during production operation.	10-146
	<ul style="list-style-type: none"> ● Overview ● Features ● Limitations ● I/O interface ● Setup ● Programming ● Testing ● Error Recovery Manual Function ● I/O timing sequence 	10-146 10-149 10-150 10-150 10-152 10-158 10-161 10-161 10-164
Auto Normal Utility (option)	Tool location and orientation are critical to cut quality in the laser shape cutting process.	10-169
	<ul style="list-style-type: none"> ● Overview ● Setup ● Execution 	10-170 10-173 10-175
Tool Center Point Speed Prediction (option)	The TCP Speed Prediction option provides the predicted value of the TCP speed continuously as the robot is moving.	10-176
	<ul style="list-style-type: none"> ● Overview ● System Variables 	10-177 10-179
Program ToolBox (option)	The following features make up the Program ToolBox option:	10-184
	<ul style="list-style-type: none"> ● Cross Car Mirror (option) ● UTOOL Adjust (option) ● Flip Knuckle (option) ● Limit Set (option) 	10-185 10-189 10-194 10-197

Advanced functions provide extended capability to your system.

Advanced functions include:

NOTE You will be able to use the advanced functions described in this chapter only if you have purchased them and they have been installed properly. Not all of the functions described in this chapter are available in North America.

10.1 MIRROR IMAGE UTILITY

The mirror image utility allows for translating 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, or to copy a program used on a left-hand robot to a right-hand robot.

Mirror imaging of a program can be accomplished either as a

- Parallel mirror image
- Parallel and rotational mirror image

Parallel Mirror Image

A parallel mirror image mirrors the program about a mirror plane without an offset or a rotation. See Figure 10-1 and Figure 10-2.

Figure 10-1. Parallel Mirror Image with Mirror Plane in Center of Robot

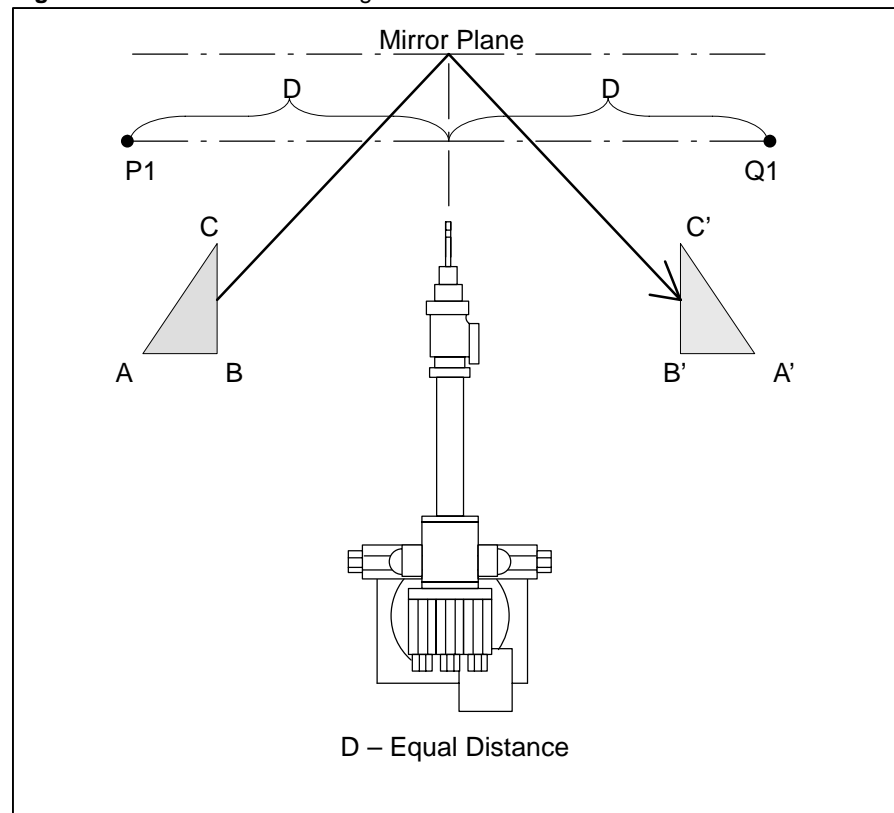
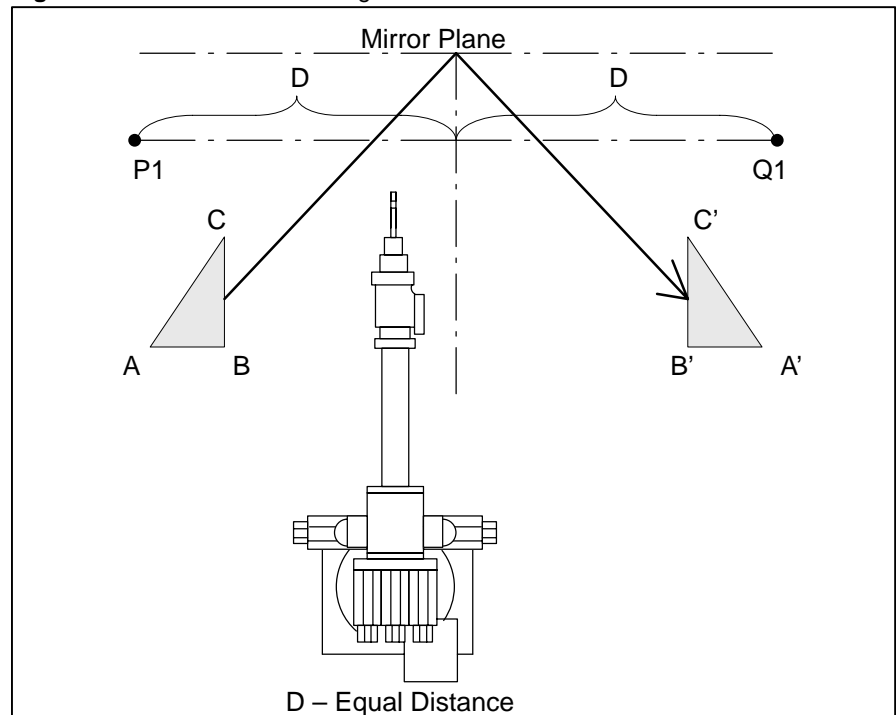


Figure 10-2. Parallel Mirror Image with Mirror Plane Offset from Center of Robot



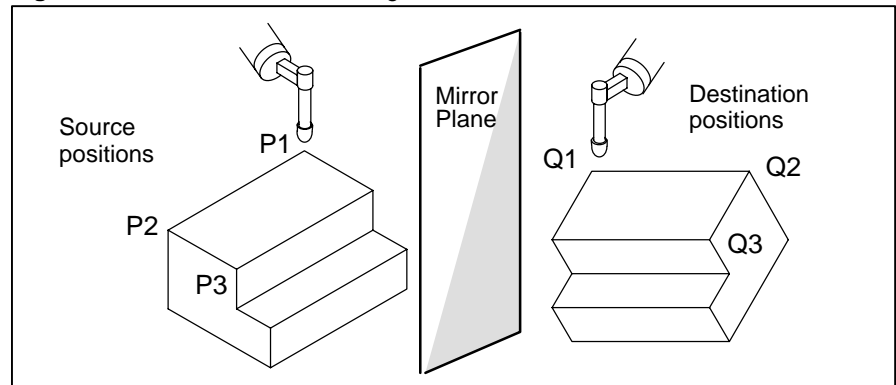
CAUTION

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.

Rotational Mirror Image

A parallel mirror image mirrors and rotates the program about a mirror plane. Orientation of the part to be imaged is rotated about one or more of its axes relative to the mirror plane. See Figure 10-3.

Figure 10-3. Rotational Mirror Image

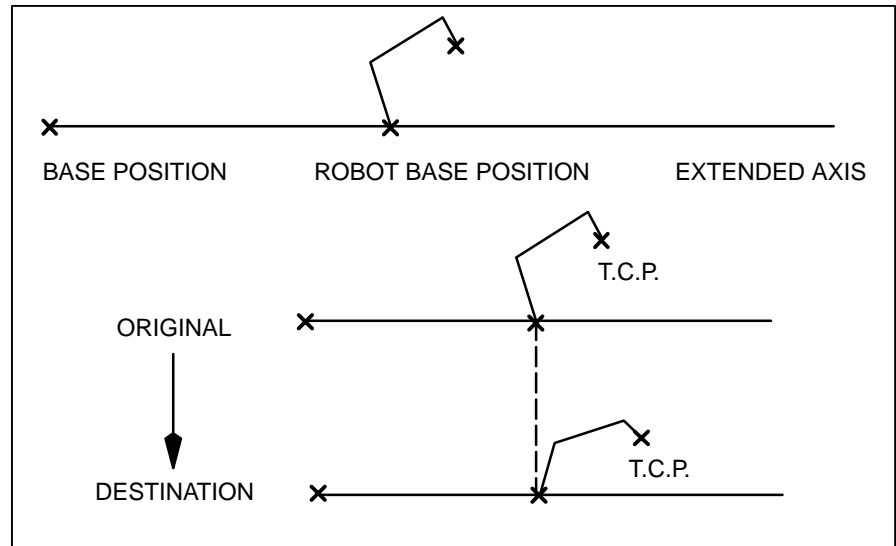


Mirror Image of Extended Axes

Extended (Ext) axes determines how the mirror image function will translate the program when you are using extended axes. Possible extended axes configurations are:

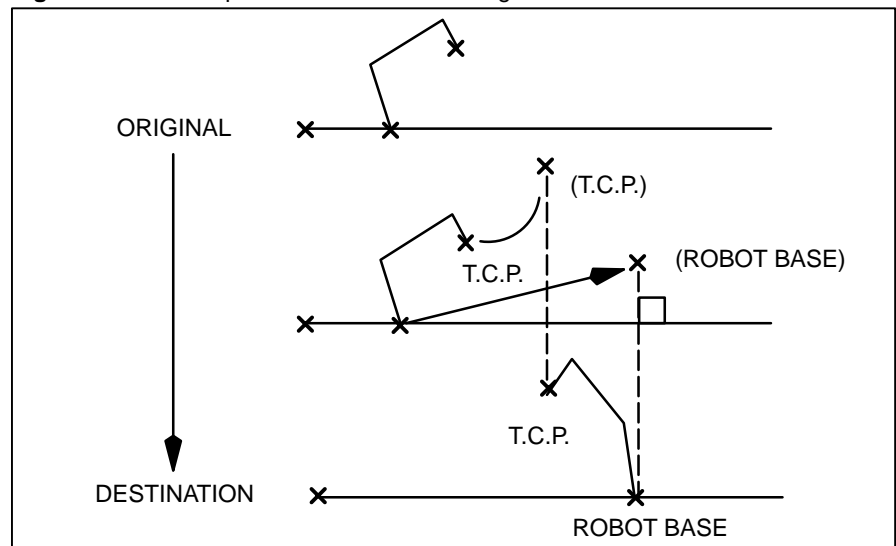
- **Robot axes only** – allows you to mirror the axes of the robot without mirroring any non-integrated extended axes such as a positioning table. The shift is calculated using the change in the robot tool center point (TCP). See Figure 10-4.

Figure 10-4. Example of Robot Axes Only Shift



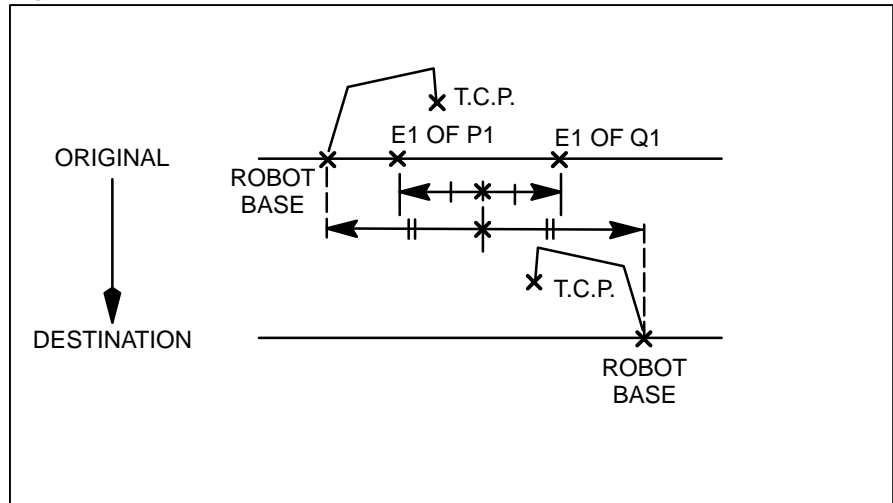
- **Ext integrated** – allows you to mirror the axes of the robot and any integrated axes. The amount of mirror image for the robot and the extended axes is calculated using the change in the TCP. See Figure 10-5.

Figure 10-5. Example of Extended Axes Integrated Shift



- **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 Figure 10-6.

Figure 10-6. Example of With Extended Axes Shift



- **Ext axes only** – Ext axes only is not available for a mirror shift.
- **Replace Ext axes** – Replace Ext axes is not available for a mirror shift.

Use Procedure 10-1 to perform a mirror image of a program.

Procedure 10-1 Using Mirror Image

- | | |
|------------------|---|
| Condition | <ul style="list-style-type: none"> ■ The program you want to mirror has been created and contains recorded positions. ■ If you are using mirror image to transfer positions between right and left-hand robots, be sure you are performing the mirror image on the destination robot and not the source robot. Therefore, you must first copy the program from the source robot to the destination robot before you perform the mirror image. ■ All robot joint axes are at zero degrees. |
| Step | <ol style="list-style-type: none"> 1 Press MENUS. 2 Select UTILITIES. 3 Press F1, [TYPE]. 4 Select Mirror Image. You will see the Mirror Image Shift screen. |

- 5 Move the cursor to Original Program. If the program you want to mirror is not selected, press ENTER. Use the appropriate function keys to type the name of the program 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 and select to mirror the WHOLE program or PART of the program.
 - **To mirror the whole program, press F5, WHOLE.** You will see a screen similar to the following.

```

MIRROR IMAGE SHIFT          JOINT  50 %
                               2/7
PROGRAM
1 Original Program:          [STYLE37]
2   Range:                   WHOLE
3   Start line: (not used)   ****
4   End line: (not used)    ****
5 New Program:              [      ]
6   Insert line: (not used)  ****
7 EXT axes :                 Robot axes only

To move page with SHIFT + DOWN, SHIFT + UP

[ TYPE ]                     PART  WHOLE >

```

NOTE EXT axes will only be displayed 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          JOINT  50 %
                               2/7
PROGRAM
1 Original Program:          [STYLE37]
2   Range:                   PART
3   Start line:              0
4   End line:                0
5 New Program:              [      ]
6   Insert line: (not used)  ****
7 EXT axes :                 Robot axes only

To move page with SHIFT + DOWN, SHIFT + UP

[ TYPE ]                     PART  WHOLE

```

7 If you selected to mirror PART of a program,

- 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 Figure 10-7 and Figure 10-8.

Figure 10-7. Mirroring an Entire Program

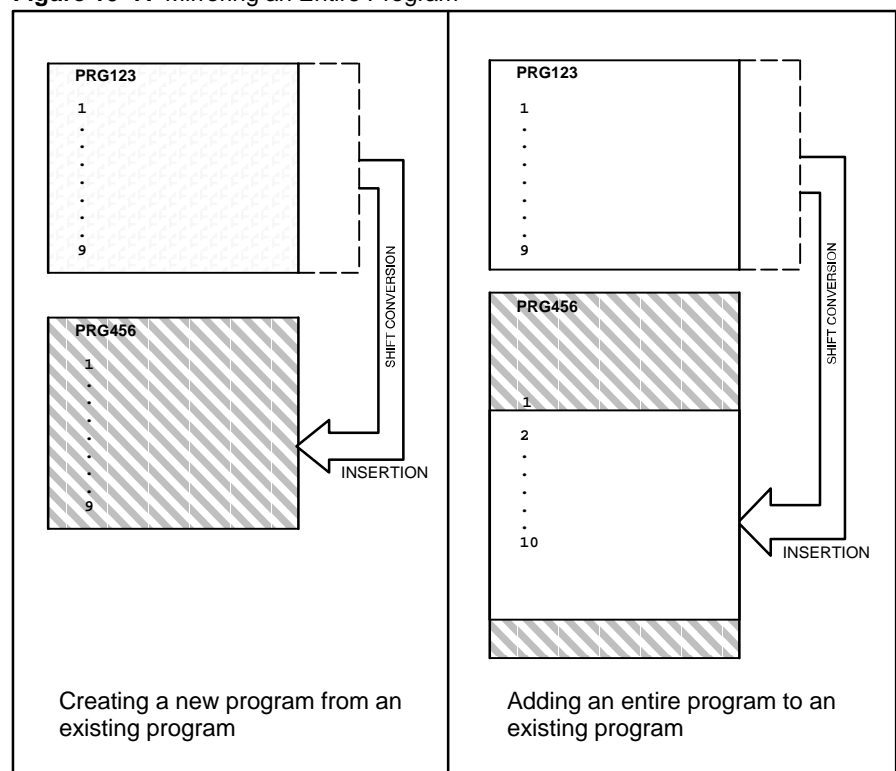
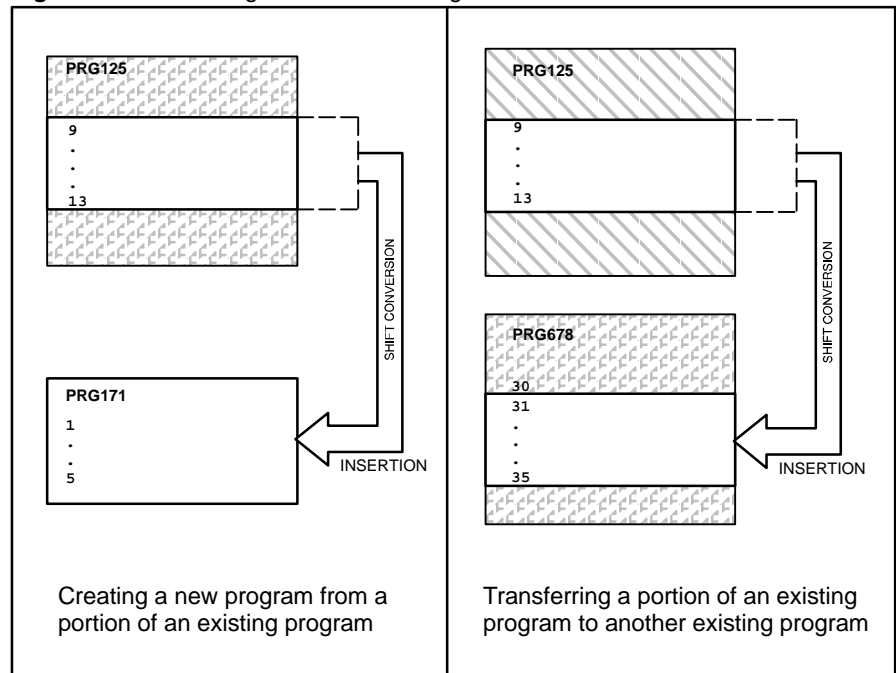
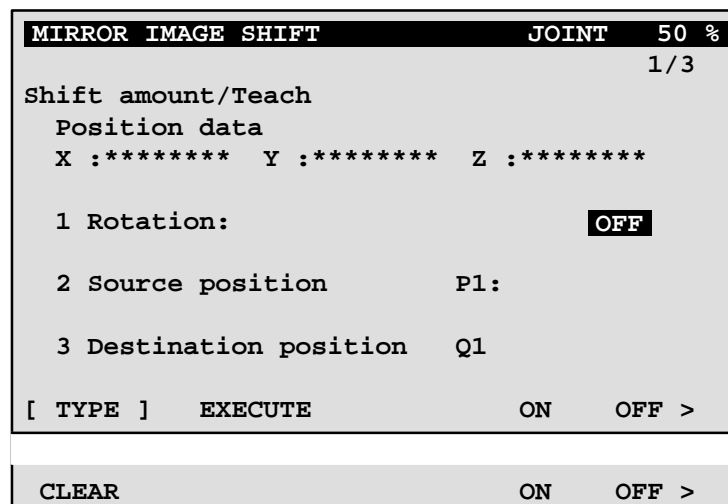


Figure 10-8. Mirroring a Portion of a Program



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 data.
- 9 Press the down arrow key. You will see the Mirror Image Shift (Position) screen.
 - 10 Move the cursor to rotation. You will see a screen similar to the following.



HINTS

P1 and Q1 should not be part of the program to be mirrored. They should be new positions that are located at an equal distance from the mirror plane. (See Figure 10-1.) Use the following guidelines to teach these positions:

- a** Jog the robot to zero degrees by matching up the witness marks on the robot (if they are available), or by displaying the POSITION screen.
- b** Jog the robot:
 - Set the jog coordinate system to WORLD.
 - Jog the robot in +Y by a known distance, 200 mm for example.

OR

- Set the jog coordinate system to JOINT.
- Jog the robot a known angle, 20°, for example.
- c** Record this position as P1.
- d** Jog the robot back to zero.
- e** Jog the robot in the opposite direction the exact distance or from the mirror plane as you jogged the robot in Step b.
- f** Record this position as Q1. See Figure 10-1.

11 Decide whether or not you want to rotate the positions.

Not Rotating the Positions

- **If you do not want to rotate the positions**, press F5, OFF. You will see a screen similar to the following.

MIRROR IMAGE SHIFT		JOINT	50 %
			1/7
SHIFT AMOUNT/TEACH			
Position data			
X :*****	Y :*****	Z :*****	
1 Rotation:			OFF
2 Source position		P1:	
3 Destination position		Q1	
[TYPE]	EXECUTE	ON	OFF >
CLEAR		ON	OFF >

- a Move the cursor to Source position (see Figure 10-1). You will see a screen similar to the following.

MIRROR IMAGE SHIFT		JOINT	50 %
			2/7
SHIFT AMOUNT/TEACH			
Position data			
X :*****	Y :*****	Z :*****	
1 Rotation:			OFF
2 Source position		P1:	
3 Destination position		Q1	
[TYPE]	EXECUTE	REFER	RECORD >
CLEAR			>

- b 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 the 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.
- c Move the cursor to Destination position (see Figure 10-1). You will see a screen similar to the following.

Input position register number:
P[] PR[]

MIRROR IMAGE SHIFT		JOINT	50 %
			1/7
MIRROR IMAGE SHIFT (POSITION)			
Position data			
X :*****	Y :*****	Z :*****	
1 Rotation:			OFF
2 Source position		P1: P[1]	
3 Destination position		Q1:	
[TYPE]	EXECUTE	REFER	RECORD >
CLEAR			>

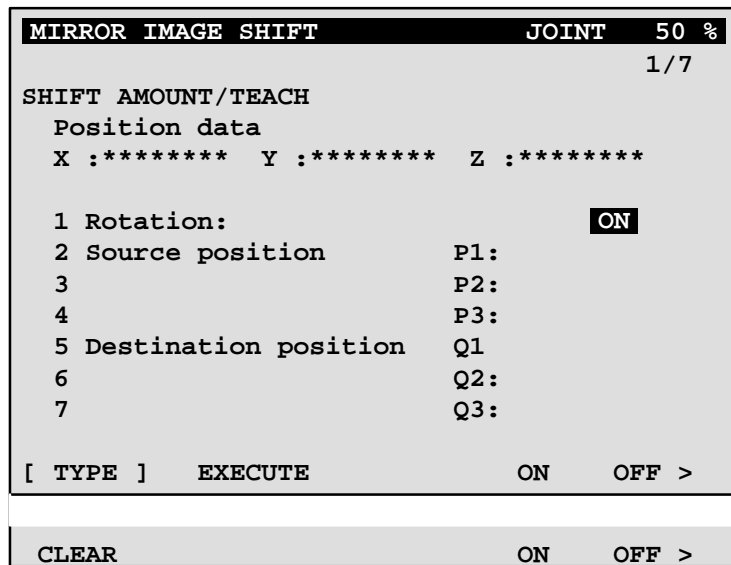
Input position register number:
P[] PR[]

- **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 or position register**, press F4, REFER. Select the position or position register.

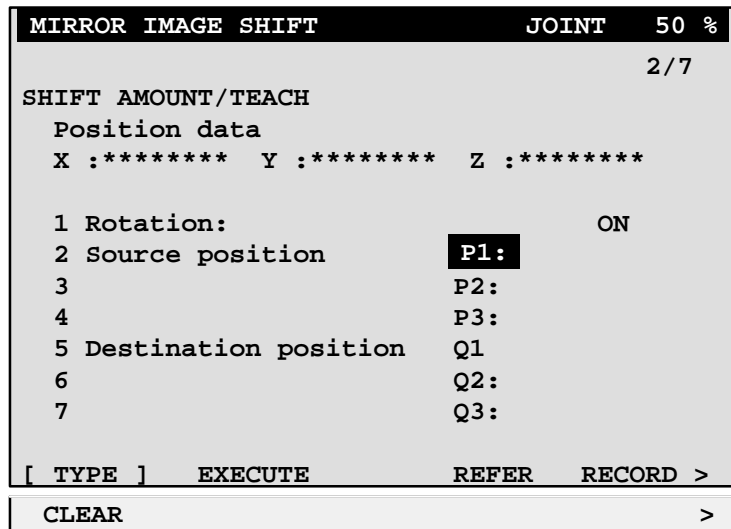
d Record or specify the destination position.

Rotating the Positions

- **If you want to rotate the positions**, press F4, ON. You will see a screen similar to the following.

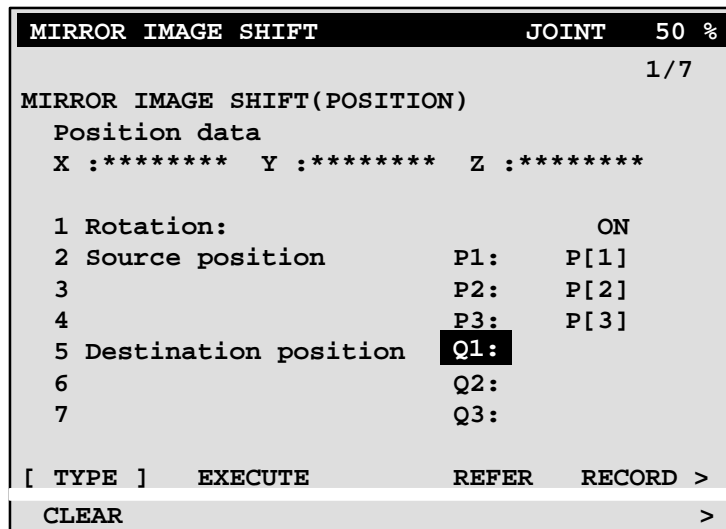


- a Move the cursor to Source position (see Figure 10-3). You will see a screen similar to the following.



Input position register number:
P[] PR[]

- b 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 the 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.
- c Record or specify all the source positions.
- d Move the cursor to Destination position (see Figure 10-3). You will see a screen similar to the following.



Input position register number:
P[] PR[]

- **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 or position register**, press F4, REFER. Select the position or position register.
- e 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.

Execute Transform ok?

- 12 **To mirror image the program**, press F2, EXECUTE.
 - **To execute the mirror image shift**, press F4, YES.
 - **To not execute the mirror image shift**, press F5, NO.

-PROCESSING-

- 13 Wait until software has finished processing the mirror image.

Troubleshooting

Some positions in your program might not be able to be mirrored.

When this happens, the mirror image software keeps the position in the program at the exact location and orientation it was in before the mirror image was executed. To correct this, you must reattach the position manually.

If this happens while your mirror image program is processing, you will see a message similar to the following.

Select P[1] : J6 angle (deg -234)				
deg -234	deg 126	*Uninit*	QUIT	
F1	F2	F3	F4	F5

This message will be displayed for each position that cannot be mirrored. Note the position number; P[1] in this example.

To continue, usually press F1. This will allow the mirror image program to continue processing. Continue noting the position number and axis for each position that did not mirror correctly.



WARNING

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 manually reattach each position that did not mirror correctly.

10.2 PROGRAM SHIFT UTILITY

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.

Shifting a program can be accomplished either as a

- Parallel shift
- Parallel and rotational shift

You can shift the positions of an entire program or a portion of a program from within a program or from one program to another. In this way, robot paths can be transferred from one program to another or one robot to another in order to perform backups. See Figure 10-9 and Figure 10-10.

Figure 10-9. Shifting an Entire Program

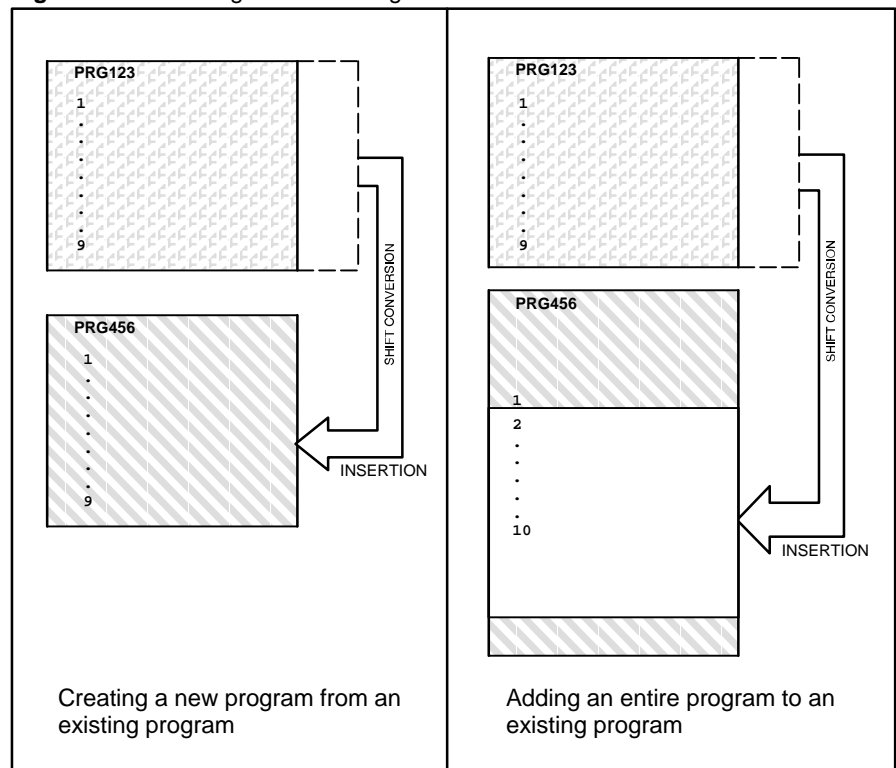
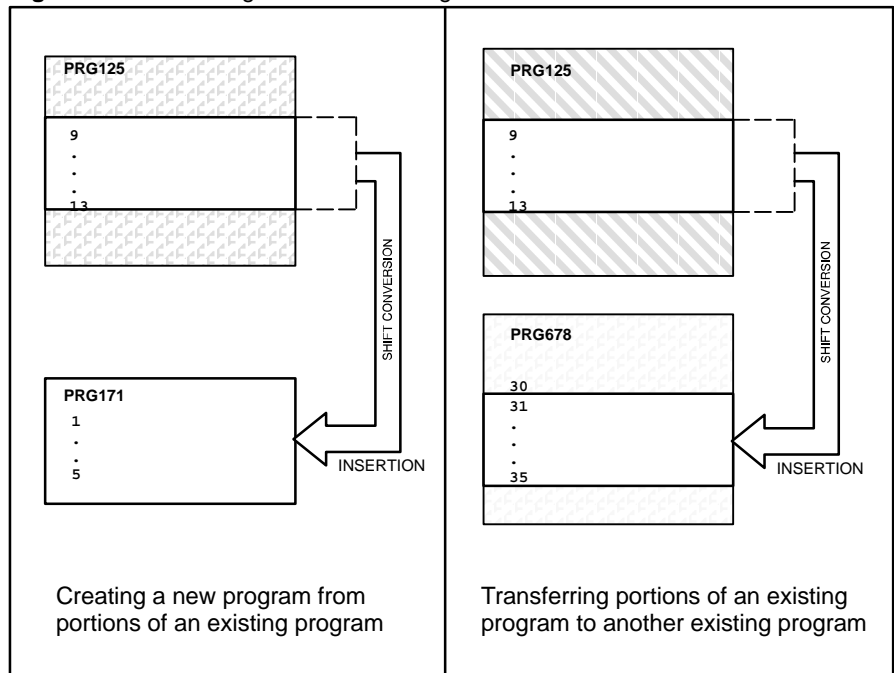


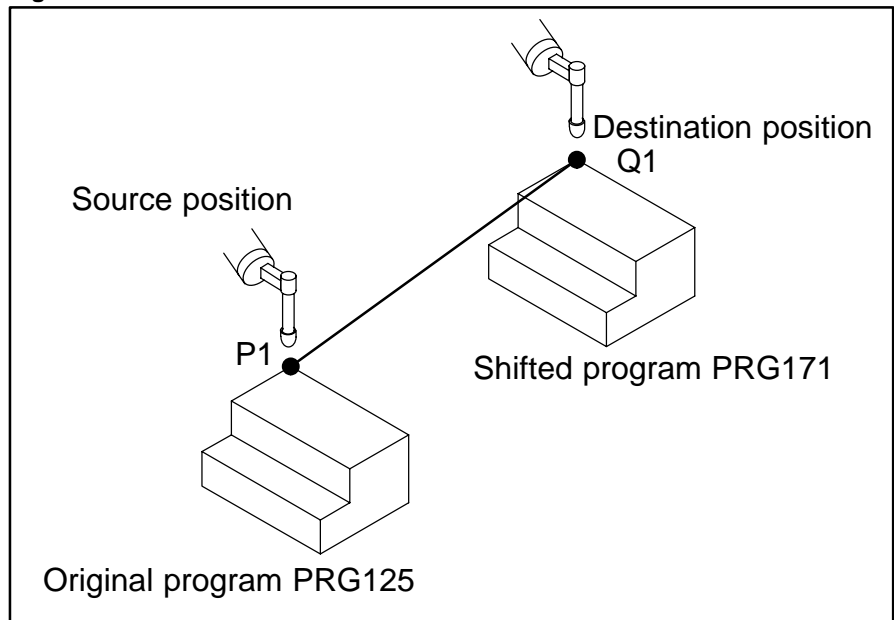
Figure 10-10. Shifting Portions of a Program



Parallel Shift

A parallel shift of a program is accomplished by reteaching the location of one point from the original (source) program, to the destination program. See Figure 10-11.

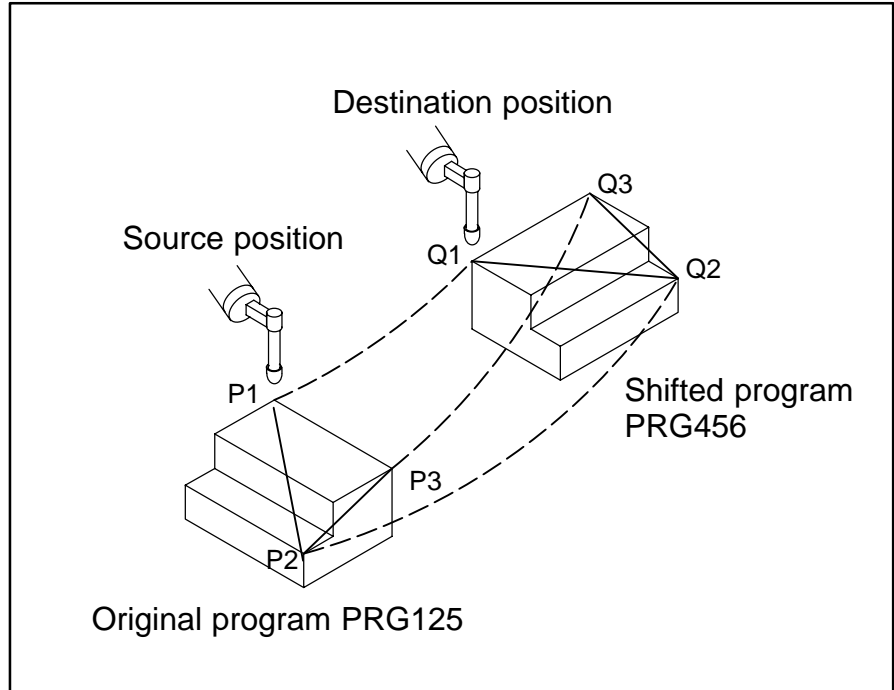
Figure 10-11. Parallel Shift



Parallel and Rotational Shift

A parallel and rotational shift is accomplished by reteaching the location of three points from the original (source) program (P1, P2 and P3) to the destination program (Q1, Q2 and Q3). See Figure 10-12.

Figure 10-12. Parallel and Rotating Shift

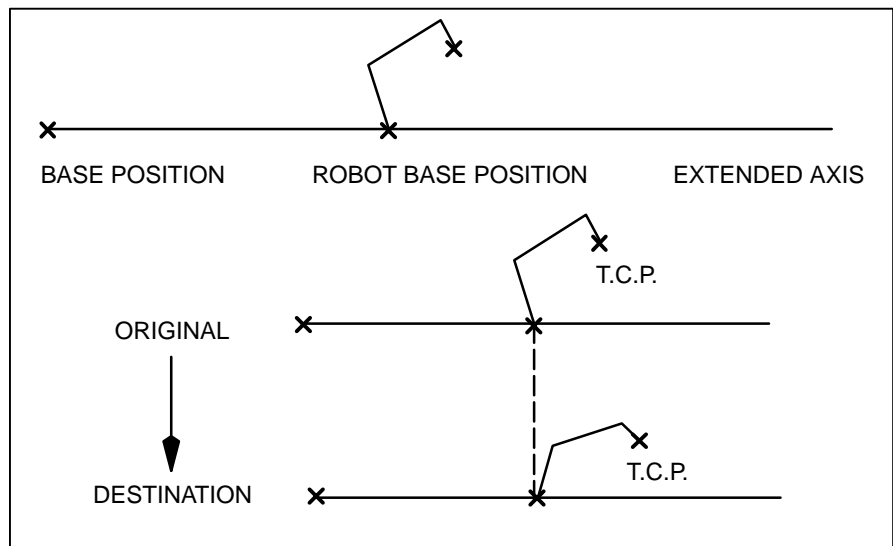


Extended Axes

Extended (Ext) axes determines how the program shift function will translate the program when you are using extended axes. Possible ext axes types are:

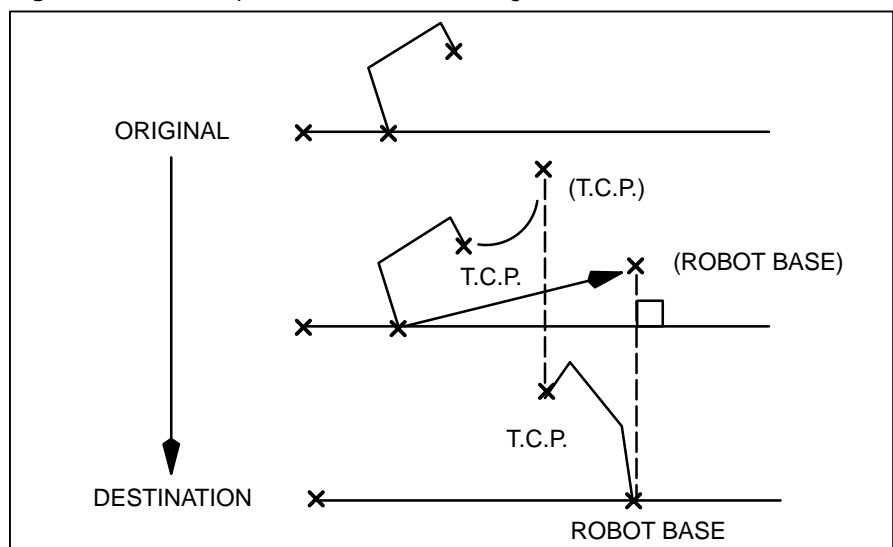
- Robot axes only** – allows you to shift the axes of the robot without shifting a non-integrated extended axes such as a positioning table. The shift is calculated using the change in the robot tool center point (TCP). See Figure 10-13.

Figure 10-13. Example of Robot Axes Only Shift



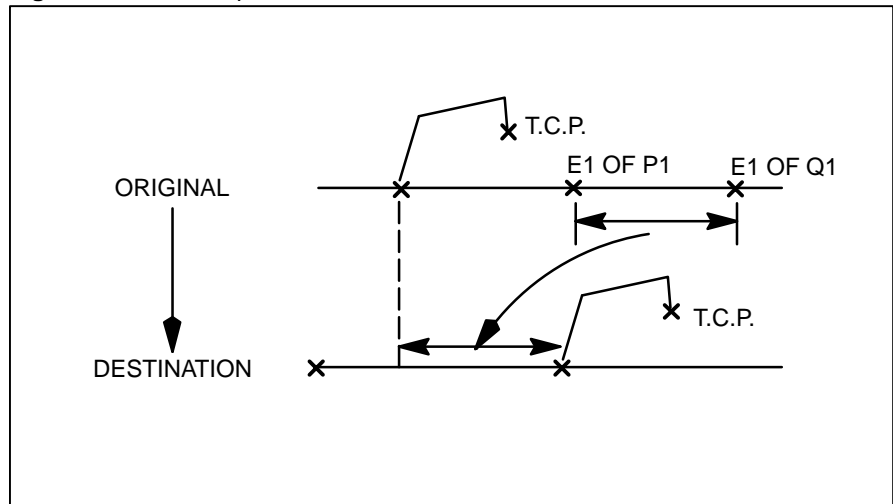
- Ext integrated** – allows you to shift the axes of the robot and any integrated axes. The amount of shift for the robot and the extended axes is calculated using the change in the TCP. See Figure 10-14.

Figure 10-14. Example of Extended Axes Integrated Shift



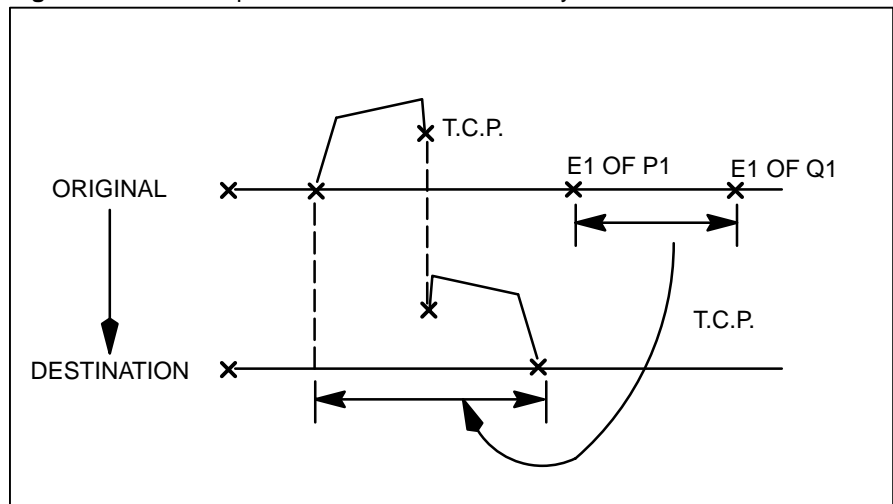
- **With ext axes** – allows you to shift positions for robot axes and any extended axes in your system. The amount of shift for the robot is calculated by using the change in the TCP. The amount of shift for the extended axes is calculated using the difference between an original position (P1) and a new position (Q1). See Figure 10-15.

Figure 10-15. Example of With Extended Axes Shift



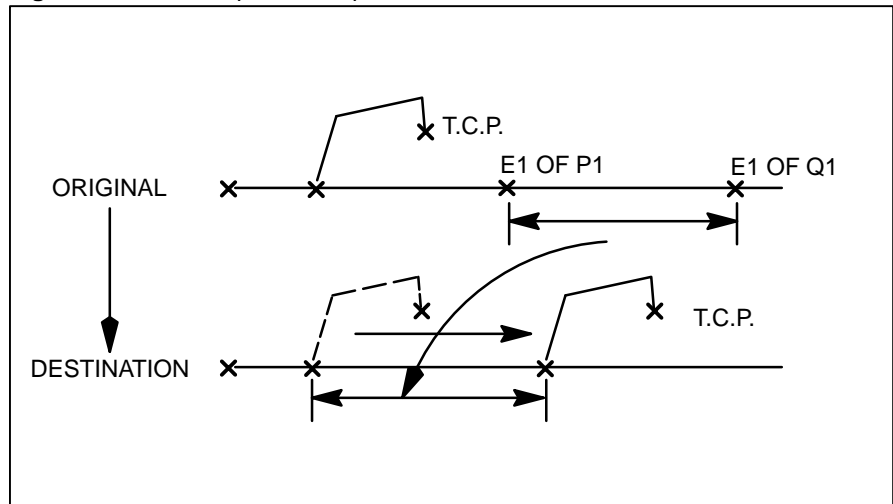
- **Ext axes only** – allows you to shift positions for the extended axes in your system while maintaining a constant TCP location. The amount of shift for the extended axes is calculated using the difference between an original position (P1) and a new position (Q1). See Figure 10-16.

Figure 10-16. Example of With Extended Axes Only Shift



- **Replace Ext axes** – allows you to shift positions for only the extended axes in your system without affecting any robot angles. The shift amount for the extended axes is calculated using the difference between an original position (P1) and a new position (Q1). See Figure 10-17.

Figure 10-17. Example of a Replace Extended Axes Shift



Use Procedure 10-2 to perform a program shift.

Procedure 10-2 Using the Shift Utility

- Condition** ■ The program you want to shift has been created and contains recorded positions.
- Step**
- 1 Press **MENUS**.
 - 2 Select **UTILITIES**.
 - 3 Press **F1**, [**TYPE**].
 - 4 Select **Program shift**.
 - 5 Move the cursor to **Original Program**. If the program you want to shift is not selected, press **ENTER**. Use the appropriate function keys to type the name of the program and press **ENTER**.

- 6** Move the cursor to Range and select to shift the **WHOLE** program or **PART** of the program.
- **To shift the whole program press F5, WHOLE.** You will see a screen similar to the following.

```

PROGRAM SHIFT                                JOINT  50 %
                                           2/7
PROGRAM
1 Original Program:                        [STYLE37]
2   Range:                                WHOLE
3   Start line: (not used)                 ****
4   End line:   (not used)                 ****
5 New Program:                             [    ]
6   Insert line: (not used)                 ****
7 EXT axes :                               Robot axes only

To move page with SHIFT + DOWN, SHIFT + UP

[ TYPE ]                                PART  WHOLE

```

- **To shift part of the program press F4, PART.** You will see a screen similar to the following.

```

PROGRAM SHIFT                                JOINT  50 %
                                           2/7
PROGRAM
1 Original Program:                        [STYLE37]
2   Range:                                PART
3   Start line:                            0
4   End line:                               0
5 New Program:                             [    ]
6   Insert line:                            ****

To move page with SHIFT + DOWN, SHIFT + UP

[ TYPE ]                                PART  WHOLE

```

7 If you selected to shift **PART** of a program,

- Move the cursor to Start line and type the starting line number.
- Move the cursor to End line and type the ending line number.

- 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 shifting the positions.

This can be the same as the original program, an already existing program, or a program that does not exist.

- 9 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 data.
- 10 Press the SHIFT key while pressing the down arrow key. You will see the Shift Position screen.
- 11 Move the cursor to rotation. You will see a screen similar to the following.

PROGRAM SHIFT		JOINT	50 %
			1/3
SHIFT AMOUNT/TEACH			
Position data			
X :*****	Y :*****	Z :*****	
1 Rotation:			OFF
2 Source position		P1:	
3 Destination position		Q1	
[TYPE]	EXECUTE	ON	OFF

- To rotate the positions, press F4, ON. You will see a screen similar to the following.

PROGRAM SHIFT		JOINT	50 %
			1/7
SHIFT AMOUNT/TEACH			
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

- 12 Move the cursor to Source position (see Figure 10-11 and Figure 10-12). See the following screen for an example.

PROGRAM SHIFT		JOINT	50 %
			2/7
Shift amount/Teach			
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 >
CLEAR			>

- 13 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 the SHIFT key and press F5, RECORD.
- **To specify a previously recorded position or position register**, press F4, REFER. Select the position or position register, and press ENTER.

Input position register number:
P[] PR[]

- 14 **If you are rotating the positions**, record or specify all the source positions.

- 15 Move the cursor to Destination position (see Figure 10-11 and Figure 10-12). See the following screen for an example.

PROGRAM SHIFT		JOINT	50 %
			5/7
PROGRAM SHIFT (POSITION)			
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 >
CLEAR			>

- **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 or position register**, press F4, REFER. Select the position or position register.

Input position register number:
P[] PR[]

16 If you are rotating the positions, record or specify all the destination positions.

Execute Transform ok?

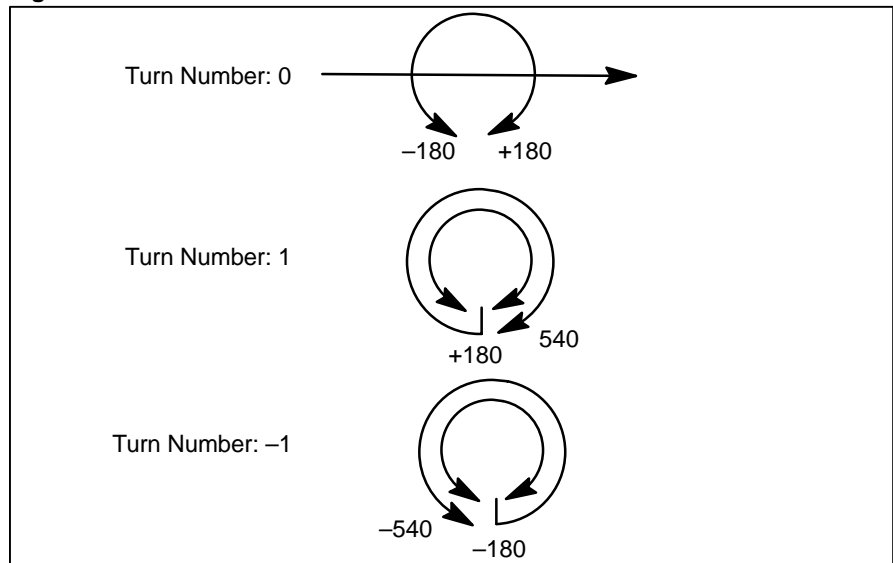
17 To shift the program, press F2, EXECUTE.

- To execute the shift, press F4, YES.
- Not to execute the shift, press F5, NO.

NOTE For each position to shift, you receive the message “Select P[n]: m angle (deg p)” (where n equals a position number of the destination program and m equals the amount of angle) only if the shift causes:

- A joint to wrap greater than 180 degrees.
- The turn number of the joint to change if the wrap is less than 180 degrees. See Figure 10-18.

Figure 10-18. Turn Numbers



Select P[1]: J6 angle(deg p)
deg p deg q *uint* QUIT

18 Select the type of angle to use.

- To allow joints to wrap and/or turn number to change, press F1, deg p. Normally, the angle change will be less than 180 degrees, but will have a different turn number.
- To not allow any wrapping or a change in turn number, press F2, deg q. The angle change will be greater than 180 degrees, but the turn number is the same.
- To not allow any wrapping, a change in turn number, and any angle changes, press F3, *uint*. The joint angles for that position will remain uninitialized. You will have to reattach the position after the transformation has completed.
- To stop the shift for each position, press F5, QUIT.

-PROCESSING-

19 Wait until software has finished processing the shift.

10.3 SPACE CHECK FUNCTION

The space check function, incorporated into a robot, monitors a predetermined interference area (space). When another robot or a peripheral unit is located within that space, the function stops robot operation if a move command specifying movement into that space is issued to the robot. The space check function releases the stop state and allows robot operation to continue only after checking that the other robot or peripheral unit has moved out of the area.

Two interlock signals are assigned to a single interference area: one input and one output. These interlock signals are used for communication between a peripheral unit and the robot. You can define up to three interference areas.

Interlock Output Signal

When the tool center point enters the interference area, the interlock output signal goes off. When the tool center point is not located within the area, the signal is on. Refer to Table 10-1.

Table 10-1. Interlock Output Signal Operation

Condition	Output
Safe (the tool is not located within the area)	ON
Dangerous (the tool is located within the area)	OFF

Interlock Input Signal

When an attempt is made to move the robot into the interference area while the interlock input signal is off, the robot enters a hold state. When the input signal goes on, the hold state is released and the robot resumes automatic operation.



WARNING

Since the robot starts decelerating as soon as the tool center point enters the interference area, the robot can stop at a point within the area. The faster the robot speed, the further within the area is the point where the robot stops. Taking this fact and the size of the tool into consideration, specify a larger interference area than actually exists.

You set up the space check function using the SETUP Space Fnct screens. Table 10-2 lists and describes the items on the Space Fnct screens.

Table 10-2. Space Check Function Screen Items

ITEM	DESCRIPTION
Rectangular Space DETAILED Screen Items	
Enable/Disable	Enables and disables the space check function. To set or modify other conditions for an area, you must set the condition to Disable for that area.
Comment	You can specify up to ten characters as a comment.
Output Signal	Specifies the number of the interlock output signal.
Input Signal	Specifies the number of the interlock input signal.
Priority High/Low	<p>Specifies which of two robots has priority when both robots, which both use the space check function, attempt to enter the same area simultaneously. The robot for which Priority High has been specified can enter the area first. After that robot leaves the area, the other robot, for which Priority Low has been specified, is allowed to enter the area. A different setting must be made for each robot.</p> <div data-bbox="521 905 1417 1507" style="border: 1px solid black; padding: 10px;"> <p>⚠ WARNING</p> <p>If the same setting (Priority High or Priority Low) is specified for both robots and they both attempt to enter the interference area simultaneously, both enter a dead lock state. In this case, check that the correct priority has been specified for both robots and perform the following recovery:</p> <ol style="list-style-type: none"> 1. Apply emergency stop to both robots. Note that if emergency stop is not applied to both robots, as soon as one robot leaves the interference area, the other robot will start operating automatically. This is extremely dangerous. Never attempt this operation without applying emergency stop to both robots. 2. Check that the immediate vicinity of the robot is clear or personnel and equipment. 3. Disable the space check function. 4. Move one robot out of the interference area by jogging it. </div>
inside/outside	Specifies whether the inside or outside of the rectangular box you define in space is used as the interference area.
SPACE SETUP Screen Items	
BASIS VERTEX	Specifies one of the corners of the rectangular space you define as a reference.
SIDE LENGTH/ SECOND VERTEX	<p>For SIDE LENGTH, specify the length of each side of the rectangular space, relative to the reference vertex along the x-, y-, and z-axes of the user coordinate system. Each side of the box must be parallel to an axis of the user coordinate system.</p> <p>For SECOND VERTEX, specify a vertex other than the reference vertex. The rectangular space that consists of the reference vertex and the specified vertex as its diagonal vertexes is the interference area.</p>

Use Procedure 10-3 to set the conditions for the space check function.

Procedure 10-3 Setting the Conditions for the Space Check Function

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press the F1, [TYPE].
 - 4 Select Space fct. The space list screen appears. See the following screen for an example.

Rectangular Space		JOINT 10%
LIST SCREEN		1/3
No.Enb/Dsbl	Comment	Usage
1	DISABLE[AG]Common Space
2	DISABLE[]Common Space
3	DISABLE[]Common Space
[TYPE] GROUP# DETAIL ENABLE DISABLE		

- 5 Move the cursor to the condition you want to set.
- 6 **To enable or disable a condition,**
 - Press F4, ENABLE to enable it.
 - Press F5, DISABLE to disable it.
- 7 **To specify a comment,**
 - a Move the cursor to the comment space and press ENTER.
 - b Select a method of naming the comment.
 - c Press the appropriate function keys to enter the comment.
 - d When you are finished, press ENTER.
- 8 **To specify items other than Enb/Dsbl or Comments,** press F3, DETAIL. See the following screen for an example.

Rectangular Space		JOINT 10%
DETAILED SCREEN		1/6
SPACE :1	GROUP :1	
USAGE : Common Space		
1 Enable/Disable:	DISABLE	
2 Comment:	[AG]	
3 Output Signal:	DO [0]	
4 Input Signal:	DI [0]	
5 Priority:	High	
6 inside/outside:	Inside	
[TYPE] SPACE ENABLE DISABLE		

- 9 Move the cursor to the item you want to change. Use the function keys or numeric keys to enter the appropriate information.
- 10 To define the location and size of a space, press F2, SPACE. The space setting screen appears. See the following screen for an example.

Rectangular Space		JOINT 10%	
SPACE SETUP		1 / 4	
SPACE :1		GROUP :1	
UFRAME :0		UTOOL :1	
1 : BASIS\VERTEX		[SIDE LENGTH]	
2 :X 0.0	mm	0.0	mm
3 :Y 0.0	mm	0.0	mm
4 :Z 0.0	mm	0.0	mm
[TYPE]	OTHER		RECORD

- 11 Specify the reference vertex (corner) and the length of each side or the diagonal vertex (corner) using one of the following methods:
 - Method 1:** Move the cursor to X, Y, and Z, in turn, on the screen, then enter the appropriate coordinates using the numeric keys.
 - Method 2:** Move the robot to the corner of the current rectangular box in space, then record the robot's current position by pressing and holding SHIFT and then pressing F5, RECORD.
- 12 **After you have specified the interference area**, press the PREV key to return to the DETAILED SCREEN. Press the PREV key again to return to the LIST SCREEN.
- 13 **After you have set the conditions**, perform a cold start for the new settings to take effect. Refer to Appendix C.

10.4 EXECUTING MULTIPLE PROGRAMS (MULTI-TASKING)

Multi-tasking allows more than one program to run on the controller on a time-sharing basis, so that multiple programs appear to run simultaneously. The maximum number of user programs, or tasks, that can be executed simultaneously is four. The default number is one.

To increase the number of user programs that can be executed simultaneously, perform a controlled start and select the PROGRAM INIT option from the controlled start menus. Refer to Appendix C for more information on performing a controlled start.

You can execute multiple programs four ways:

- SOP cycle start
- Program number select (PNS)
- RUN program instruction
- Single step

10.4.1 Guidelines

Use the guidelines in this section when writing a program for multi-tasking and when executing multiple programs.

Writing a Program for Multi-Tasking

- Make sure all of the programs involved in the multi-tasking (up to four) each use a **different motion group**. Programs that are executed at the same time cannot use the same motion group.

You specify the motion group for a program in the program header information. Refer to Section 6.1.

- Use the **ignore pause** program attribute for programs you do not want to be paused by an error, by a command (such as the HOLD or EMERGENCY STOP button), or by enabling the teach pendant. Programs that use the ignore pause attribute **cannot** have a motion group specified. This means that these programs cannot contain any motion instructions.

For example, if you have a program that monitors I/O signals, which must execute continuously regardless of external events, you must specify the ignore pause attribute.

Ignore pause behaves differently during single step execution. Refer to Section 10.4.7 for more information.

You specify whether to use ignore pause for a program in the program header information. Refer to Section 6.1.10.

Executing Multiple Programs

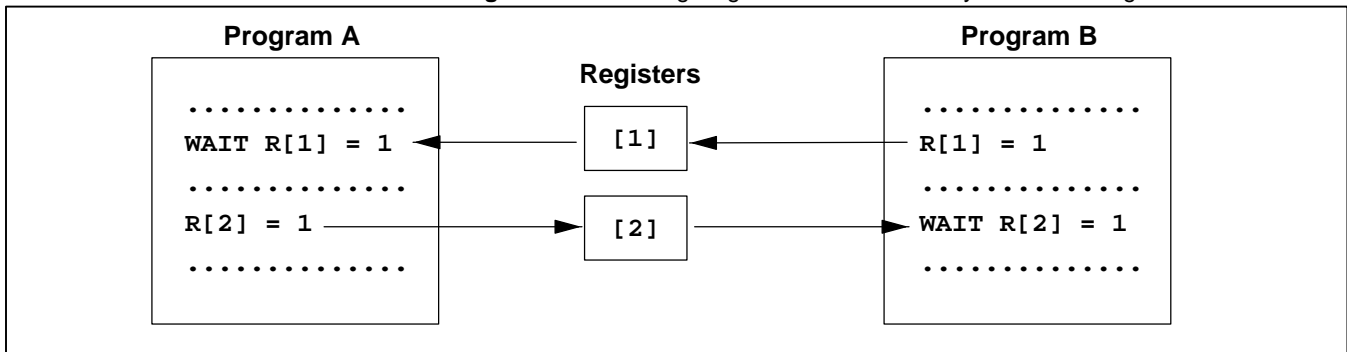
The following restrictions apply to executing multiple programs:

- Up to four programs can be run at a time.
- The controller must contain 500KB or 1MB of data memory. Refer to Section 8.12 for information on how to check the amount of data memory you have.
- You cannot execute a program if that program is currently running or paused.
- The programs you run using multi-tasking cannot use the same motion group.
- If you run a program continuously using a PLC, you must enter enough delay in the execution loop. If you do not use any delay, this program can lock other program execution.

10.4.2 Synchronizing the Execution of Multiple Programs

To synchronize the execution of two programs, use register instructions within the two programs. Figure 10-19 shows an example of register instructions used to synchronize the execution of two programs.

Figure 10-19. Using Register Instructions to Synchronize Program Execution



10.4.3 Affect of Multi-tasking on Dedicated I/O Signals

During multi-tasking program execution, keep in mind the following affects on dedicated I/O signals:

- ***IMSTP input (instantaneous stop)** is enabled for all motion groups.
- The operability of **CMDENBL output (input acceptable)** is checked for all motion groups.
- **SYSRDY output (system ready)** is checked whether servo power for all groups is supplied.

10.4.4 Standard Operator Panel (SOP) Cycle Start Execution

You can start the execution of multiple programs using the standard operator panel (SOP) CYCLE START button. Use Procedure 10-4 to execute multiple programs using the SOP CYCLE START button.

Procedure 10-4 Executing Multiple Programs Using the Standard Operator Panel (SOP) CYCLE START Button

- Condition**
- The programs you will execute at the same time do not use the same motion group.
 - You are executing no more than four programs at a time.
 - All other conditions related to executing a program using SOP CYCLE START are satisfied.
 - You are not executing the same program more than once simultaneously.
 - Mode selection switch is set to AUTO.
- Step**
- 1 Select the first program you want to execute using the SELECT menu on the teach pendant.

**WARNING**

This procedure starts production run. Make sure all safety barriers are in place, all personnel are outside of the workcell, all equipment is in place, and all production conditions have been met before you continue; otherwise, you could injure personnel or damage equipment.

- 2 Press the CYCLE START button on the operator panel.
- 3 Select the next program you want to execute using the SELECT menu on the teach pendant.

**WARNING**

This procedure starts production run. Make sure all safety barriers are in place, all personnel are outside of the workcell, all equipment is in place, and all production conditions have been met before you continue; otherwise, you could injure personnel or damage equipment.

- 4 Press the CYCLE START button on the operator panel.
- 5 Repeat Steps 3 and 4 for each program you want to execute.

10.4.5 Program Number Select (PNS) Execution

You can start the execution of multiple programs using program number select (PNS). Use Procedure 10-4 to execute multiple programs using PNS.

Procedure 10-5 Running Multiple Programs Using Program Number Select (PNS)

Condition

- The programs you will execute at the same time do not use the same motion group.
- You are executing no more than four programs at a time.
- All other conditions related to executing a program using PNS are satisfied. (Procedure 7-15)
- You are not executing the same program more than once simultaneously.
- Mode selection switch is set to AUTO.



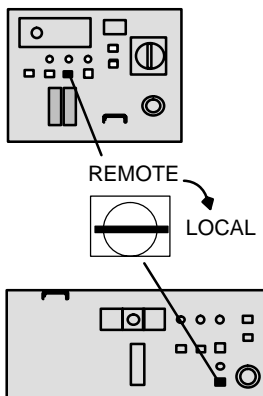
WARNING

This procedure starts production run. Make sure all safety barriers are in place, all personnel are outside of the workcell, all equipment is in place, and all production conditions have been met before you continue; otherwise, you could injure personnel or damage equipment.



WARNING

Failure to follow this procedure exactly results in the filling of the temporary memory in the R-J2 controller causing the process CPU to be locked into a busy and running condition. This could cause injury to personnel and damage to equipment. Make sure your PLC logic is correct and does not contain a high rate of production start calls.



Step

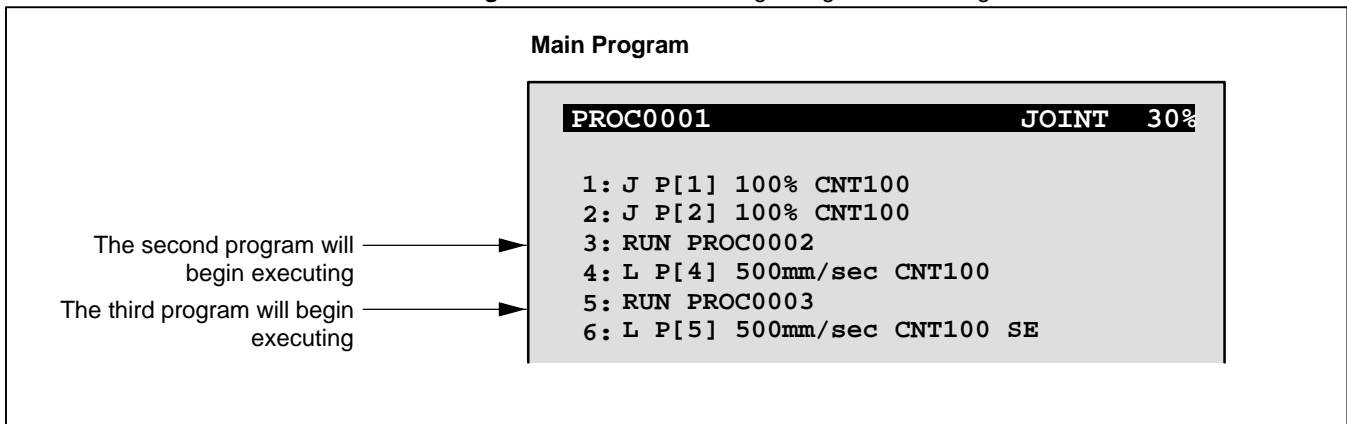
- 1 Set the LOCAL/REMOTE keyswitch on the operator panel to REMOTE.
- 2 Set the 8 bit PNS input to the number that when added to the base number will determine which program is selected. Refer to Chapter 2 for more information about PNS.
- 3 Strobe the PNSTROBE input. When the controller receives the input signal, the selected program will be displayed on the teach pendant screen. The ACK UOP signal indicates what binary input is being received. This stays ON until a new program is selected.
- 4 Press the production start button on the user operator panel to start production operation or, if your system uses a PLC, production operations will begin as soon as the PROD_START input is received.
- 5 Repeat Steps 2 through 4 for each program you want to execute.

10.4.6 RUN Program Instruction Execution

Use the RUN program within the main program to execute a second, third, or fourth program simultaneously. When you execute a program in which you have added RUN program instructions, the program you specify will execute, and execution of the main program that contains the RUN program instruction will continue at the same time.

Figure 10-20 shows an example of using a RUN program instruction to execute multiple programs.

Figure 10-20. Multi-Tasking Using the RUN Program Instruction

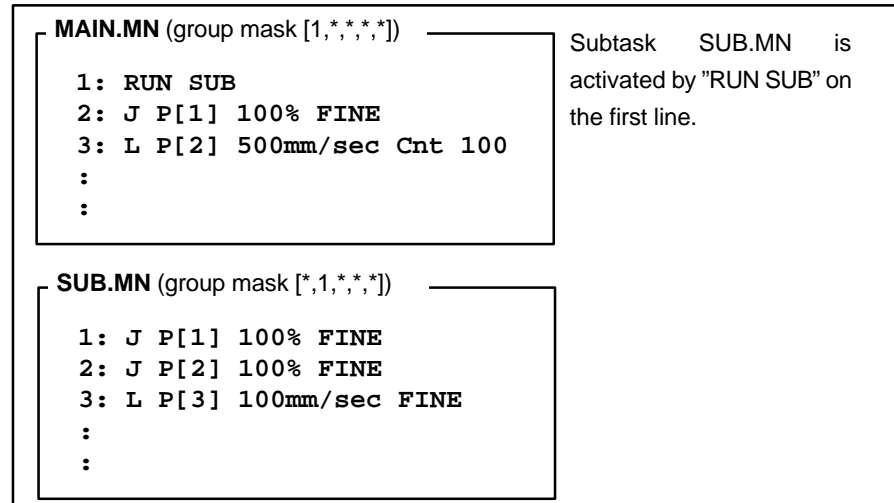


After you have included the RUN program instructions within your main program, execute the program using one of the execution methods available.

10.4.7 Single Step Program Execution

When the main task is executed in single step mode, the subtask is also executed in single step mode. A task in which a RUN instruction is issued is called a main task. A task activated by a RUN instruction within the main task is called a subtask. See Figure 10-21.

Figure 10-21. Single Step Execution Example



In Figure 10-21, the program MAIN.MN is the main task (motion group 1), and SUB.MN is the subtask (motion group 2), which is activated by a RUN program instruction within MAIN.MN.

When MAIN.MN is executed in single step mode, SUB.MN, activated by a RUN instruction in MAIN.MN, is also executed in single step mode.

Special considerations for single step execution of multi-tasking programs must also be made in the following areas:

- Ignore pause
- Backward execution
- Backward execution of the RUN instruction

Ignore Pause

When the ignore pause program attribute is ON for a program, the program is executed continuously even if single step mode has been specified for it. When you want to single step a main task, but execute a subtask continuously, set the ignore pause program attribute for the subtask to ON.

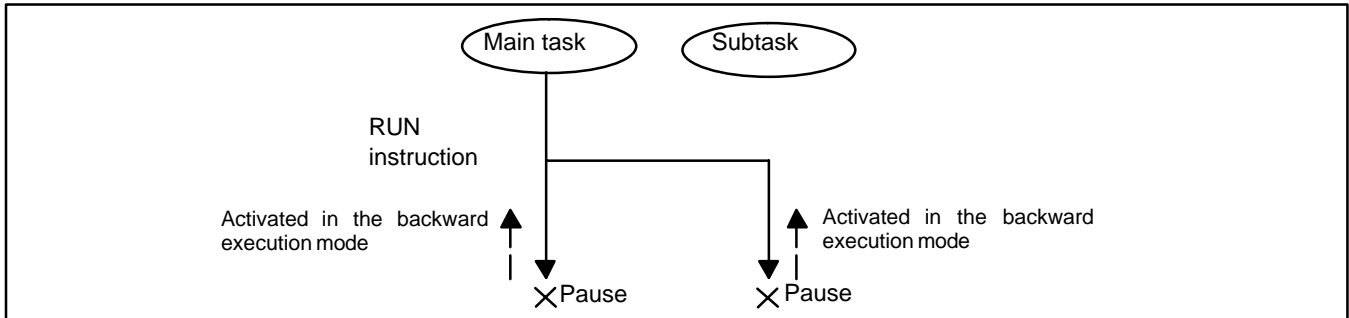
NOTE If the ignore pause program attribute is ON for a program, the program is single-stepped for safety purposes when robot motion instructions are executed. If a program does not include a robot motion instruction, set the motion mask for the program to [*,*,*,*].

The ignore pause program attribute and group mask are specified on the program detail menu.

Backward Execution

When the main task is restarted in backward execution mode, the subtask is activated also in backward execution mode. See Figure 10-22.

Figure 10-22. Single Step Backward Execution

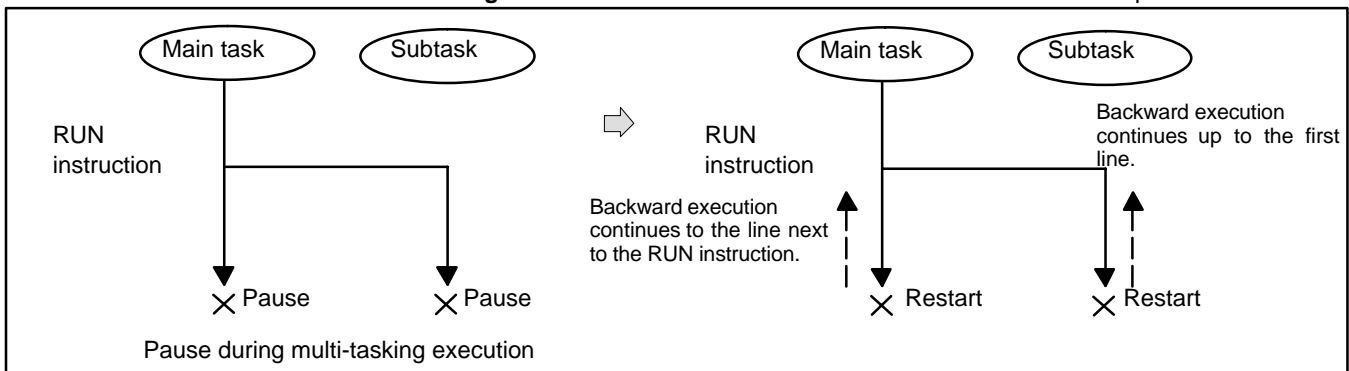


Backward Execution of a RUN Instruction

If a RUN instruction is encountered during the backward execution, it is executed in that mode. After the backward execution of the RUN instruction, the cursor moves to the instruction next to the RUN instruction. If you want to continue backward execution after the RUN instruction, move the cursor to the instruction before the RUN instruction manually, then specify the backward mode again.

In Figure 10-23, using backward execution can easily return both main task and subtask to the states they were in immediately after the RUN instruction was executed.

Figure 10-23. Backward Execution of a RUN Instruction Example



In this example, a subtask is first activated by the RUN instruction in the main task. Both tasks are then caused to pause for any reason during multi-tasking execution.

When the program is restarted from a pause during backward execution, the main task is executed to the line next to the RUN instruction in backward mode. Backward execution will not continue any further. However, the subtask is executed to the first line in backward mode. As a result:

- The main task returns to the line next to the RUN instruction.
- The subtask returns to the first line.

This is equivalent to the state immediately after the RUN instruction is executed.

10.5 ANGLE ENTRY SHIFT FUNCTION

If you are going to make multiple holes arranged at equal distances on the circumference of a circle (such as mounting bolt holes in a vehicle wheel), you can reduce the total number hours for a teach process by teaching data for one hole and having data for the other holes generated by a program shift function.

The conventional representative point teach method or the direct shift amount input method (using values in the rectangular coordinate system) is not suitable for this shift function.

This function determines the required shift amount according to three representative points taught to determine a rotation axis and directly entered rotation angles so that it can perform a programmed shift.

The angle entry shift function performs a programmed shift according to three or four representative taught points and directly entered rotation angles. Specifying a number of repetitions causes the function to make multiple shifts at equal distances on the circumference of the same circle.

Function

Transformation–source
program name
/transformation range

For the angle entry shift function, the following items must be specified.

All or part of the existing programs are specified as transformation sources. The specification method and the meaning are the same as for the program shift function.

Transformation–destination
program name
/insertion line

The angle entry shift function processes the transformation results by either of the two:

- Creating new programs
- Inserting lines into existing programs

This item is used to specify the name of a new or existing program. The specification method and the meaning are the same as for the program shift function.

Representative Points P1, P2, P3, and P0

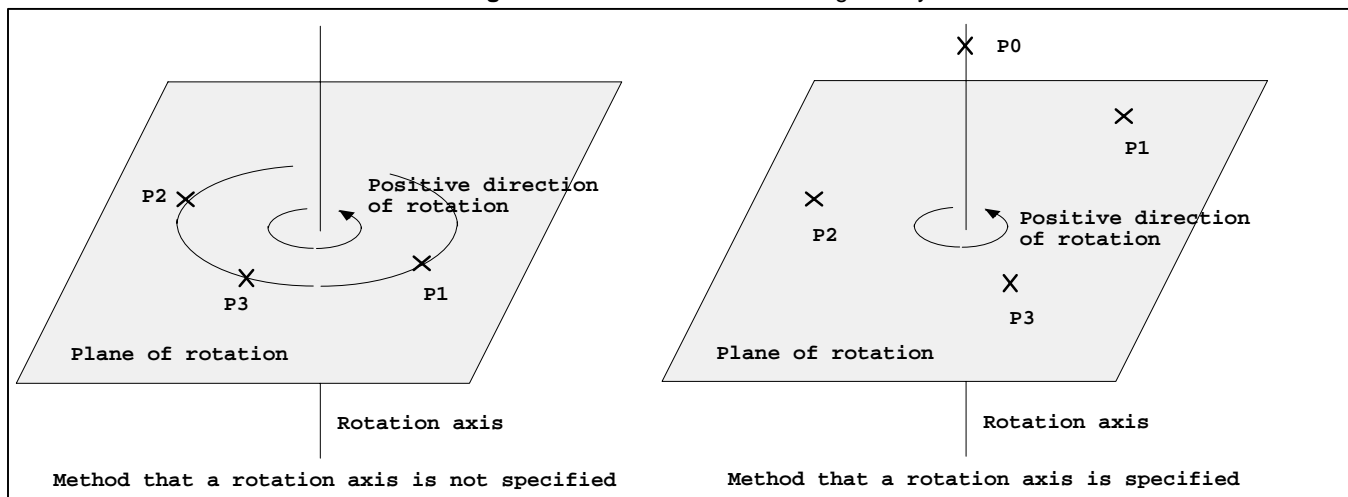
There are two methods; in one method, a rotation axis is specified and in the other method, no rotation axis is specified.

- **Method that a rotation axis is not specified (default setting)**
Three points P1, P2, and P3 are specified on the circumference of a circle. P0 is not used. The direction of rotation from P1 toward P2 is defined as positive. The plane and center of rotation are determined from these three points.
- **Method that a rotation axis is specified**
P0 is specified on the rotation axis, and P1, P2, and P3 are on the plane of rotation. The direction of rotation is defined as positive for rotation from P1 toward P2. The plane of rotation are determined from P1, P2, and P3, and an axis that is perpendicular to the plane of rotation and passes through P0 is specified as the rotation axis.

If transformation is performed without specifying a rotation axis, the calculated center of rotation is automatically specified as P0. The value for P0 can be directly changed later. So, the center of rotation can be corrected by enabling the rotation axis in the second and subsequent transformation phases.

Rotation Angle

The rotation angle is specified in degrees for the rotation center and plane determined by the three representative points. The rotation angle is directly entered as a signed real number. The positive number corresponds to the direction of rotation from P1 toward P2. See Figure 10-24.

Figure 10-24. Two Methods of Angle Entry Shift Function**Repetition Count**

The repetition count indicates how many times transformation is to be performed. If places to be operated on are arranged at equal distances on the circumference of the same circle, specifying a repetition count enables simultaneous transformation for multiple places.

If the repetition count is 2 or greater, a comment line is added at the beginning of a portion created by shifting.

Consider the following example:

Transformation-source program: program A

```
1: J P[1] 100% FINE
2: L P[2] 1500mm/sec FINE
```

If this program is transformed with the rotation angle as 20deg, the repetition count as 3, and the transformation-destination program as program B, the following program (program B) is generated with a comment added automatically.

Transformation-destination program: program B

```
1: !Angle entry shift 1 (deg 20.00)
2: J P[1] 100% FINE
3: L P[2] 1500mm/sec FINE
4: !Angle entry shift 2 (deg 40.00)
5: J P[3] 100% FINE
6: L P[4] 1500mm/sec FINE
7: !Angle entry shift 3 (deg 60.00)
8: J P[5] 100% FINE
9: L P[6] 1500mm/sec FINE
```

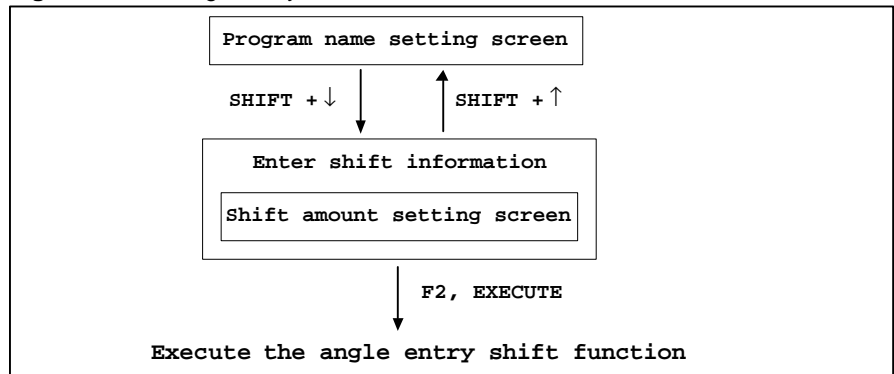
The position data for program A has been transformed as follows:

- P[1]: Position rotated by 20deg from P[1] for program A
- P[2]: Position rotated by 20deg from P[2] for program A
- P[3]: Position rotated by 40deg from P[1] for program A
- P[4]: Position rotated by 40deg from P[2] for program A
- P[5]: Position rotated by 60o from P[1] for program A
- P[6]: Position rotated by 60o from P[2] for program A

Function and Operating Procedure

The screen to set up the angle entry shift function consists of two pages; program name setting screen and shift amount setting screen. See Figure 10-25.

Figure 10-25. Angle Entry Shift Screen Structure

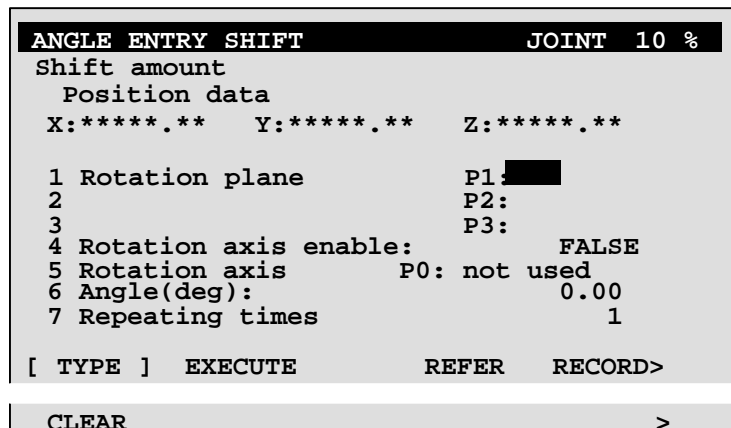


The program name setting screen works similarly to that for the program shift function. The shift amount setting screen is described below.

The angle entry shift function is set up on the UTILITIES Angle entry shift screen. First the program name setting screen appears. On this screen, specify the transformation-source program name, transformation range, transformation-destination program name, and insertion lines. (Refer to the description of the program shift utility.)

After specifying all necessary information, press the SHIFT + ↓ (down arrow) key to move to the shift amount setting screen.

The shift amount setting screen is as shown below:



Each item is specified as described below. The specification method slightly varies depending on whether to specify a rotation axis.

- Method that a rotation axis is not specified:

If “Rotation axis enable:” is set to “FALSE,” specify P1, P2, and P3, which are three points on the circumference of a circle whose center coincides with the rotation axis to be shifted. The farther these points are separated, the higher the precision of transformation.

A method to teach the representative points is the same as for the program shift function. Move the cursor to the point to be taught and press the SHIFT + F5, RECORD.

To specify a rotation angle, set the cursor on the “Angle (deg)” line and enter a signed real number representing the angle to be rotated through (a plus sign is omissible). The direction from P1 (the first point to be specified) toward P2 is assumed to be positive.

To perform transformation two or more times, set the “Repeating times” item to the desired count.

After the settings above are completed, press F2, EXECUTE. If you answer the question “Execute transform?” by pressing F4, YES, the angle entry shift function is executed.

At this point, P0 is automatically defined to be the center of the circle whose circumference coincides with P1, P2, and P3.

If a satisfactory precision is not obtained at the first transformation, change the center of the circle at the second transformation by directly specifying the rotation axis (setting “Rotation axis enable:” to “TRUE”). This method can fine-adjust transformation.

- Method that a rotation axis is specified

Setting “Rotation axis enable:” to “TRUE” causes transformation to be performed with the rotation axis specified. In this case, you specify three points in the rotation plane as P1, P2, and P3 (they need not be on the circumference of the same circle).

The teach method is the same as for method that a rotation axis is not specified. The farther these three points are separated, the higher the precision.

When this method is used, “not used” beside “P0:” disappears, indicating that P0 can be taught.

Any point on the rotational axis can be taught as P0. The accuracy of the intersection point between rotational plane and axis increases as this point gets closer to the plane defined by P1, P2 and P3.

P0 can be specified by recording position, referring position, or direct entry. The methods of recording position and referring position are the same as those for the other representative points. The direct entry method only applies to P0. Pressing the ENTER key with the cursor at the “P0” line causes the rotation axis direct entry screen, shown below, to appear.

ANGLE ENTRY SHIFT		JOINT 10 %
Shift amount		
Rotation center axis direct entry		
1 Frame:		USER FRAME 1
2 X:		0.00
3 Y:		0.00
4 Z:		0.00
[TYPE]	EXECUTE	>
CLEAR		>

On this screen, the location of P0 can be specified by entering coordinate value in an arbitrary coordinate system.

To change a coordinate system, set the cursor to the “Frame:” line and press F4, [CHOICE]. A screen appears. On this screen, select the desired coordinate system. The selected coordinate system is only used to display/specify X, Y, and Z coordinates on this screen. If the coordinate system is changed, “X:”, “Y:”, and “Z:” values are transformed to values in the new coordinate system and re-displayed.

It is also possible to directly enter values by setting the cursor to “X:”, “Y:”, or “Z:”.

If all necessary settings are completed, it is possible to execute the shift function on this screen.

When F2, EXECUTE, is pressed, the message “Execute transform?” appears. Responding to this question with F4, YES, executes the angle entry shift function.

When the shift function is executed with a rotation axis specified, the specified value is used for P0. Unlike the method with no rotation specified, no automatic calculation is performed to specify a value for P0.

To return to the shift amount setting screen (previous screen), press the PREV key.

Turn Number

In some robot models, some of the wrist axes can rotate more than 360 deg. For these robots, even if the tool center point remains at the same position, the axis angle might be in a phase different by 360 deg. In this case, if you select an incorrect angle, the axis might rotate largely when a program after transformation is executed, resulting in a dangerous behavior of the robot.

The shift function performs optimization to minimize such danger. The optimization is not necessarily perfect, however, and the following message is output together with the choices of axis angles in some cases.

```
Repeat 3: Select P[1]:J6.(deg 183)
183° -177° *uninit* QUIT >
```

- Normally select the value assigned to the F1 key ("183°" in the above example), because this is determined through optimization by the shift function.
- The value assigned to the F2 key is a value before optimization. This value is not normally used. It can be used if the value at the F1 key fails.
- Press F3, *uninit*, to cancel the taught data. If the transformation–destination program is executed under this condition, an error occurs. The data should be taught again.
- Press F5, QUIT, to abort transformation.

If a repetition count is specified for the angle entry shift function, optimization selects an angle close to the result of the previous transformation.

If the angle entry shift function performs one rotation's worth of transformation, an axis is rotated by 360°. This will generate data corresponding to a position beyond a stroke limit in many cases. To solve this problem, take either of the following actions:

- Design the transformation–source program so that the stroke limit will be not exceeded even after one rotation. This method is preferred.
- Invert the sign of the rotation angle and regenerate the position data so that it will not reach the stroke limit. In this case, two programs are generated and they have a different rotation direction. These programs must be combined later by editing. Use this method as required.

10.6 REFERENCE POSITION UTILITY

A reference position defines position limits within which an output signal will turn on. To use the reference position utility you specify a joint position for each axis of your robot, assign an output signal to the position, and, optionally, specify tolerance ranges for each axis joint position.

When the robot moves to within the tolerance range of the specified reference position joint axis locations, the assigned digital output (DO) or robot output (RO) signal turns on. When the robot moves out of the tolerance range, the DO or RO signal turns off. If a tolerance range is not specified, every axis of the robot must be at the exact reference position joint axis location for the signal to turn on.

To use a reference position, your program must contain a taught joint position whose axes locations match those of the reference position.

There are two screens associated with the reference position utility: the **LISTING** screen and the **DETAIL** screen. The listing screen allows you to view limited information for all reference positions. The detail screen allows you to view all information for a single reference position.

Table 10-3 lists and describes each item on the LISTING screen.

Table 10-4 lists and describes each item on the DETAIL screen.

Table 10-3. Reference Position LISTING Screen Items

ITEM	DESCRIPTION
No.	Shows the number of available reference positions. Three is the maximum.
Enb/Dsbl	<p>When set to Enb, allows the system to check whether the robot is at the specific joint axes positions when no tolerance range is set, or within the specified tolerance range of the joint reference position. If the robot is in the range, the specified signal is turned on. If the robot goes out of the range, the signal turns off. This can be set from the LIST screen or the DETAIL screen.</p> <p>When set to Dsbl, allows the system to ignore the reference position check. This can be set from the LIST screen or the DETAIL screen.</p>
@Pos	Indicates whether the robot is currently at any reference position.
Comment	Allows you to enter a comment about the reference position. This can be set from the LIST screen or the DETAIL screen.

Table 10-4. Reference Position DETAIL Screen Items

ITEM	DESCRIPTION
Reference position number	Indicates the reference position number the screen is currently displaying.
Comment	Allows you to enter a comment about the reference position. This can be set from the LIST screen or the DETAIL screen.
Enable/Disable	<p>When set to Enable, allows the system to check whether the robot is at the specific joint axes positions when no tolerance range is set, or within the specified tolerance range of the joint reference position. If the robot is in the range, the specified signal is turned on. If the robot goes out of the range, the signal turns off. This can be set from the LIST screen or the DETAIL screen.</p> <p>When set to Disable, allows the system to ignore the reference position check. This can be set from the LIST screen or the DETAIL screen.</p>
Signal Definition	Allows you to specify the digital output or robot output signal that turns on and off as the robot moves in and out of the specified tolerance range.
J1 through J[n]	Allows you to enter the angle of each joint for your robot that together will form the reference position.
+/- 0.000	Allows you to specify the acceptable position limits, or tolerance range, of each joint. Once the robot is within these position limits, the assigned digital output (DO) or robot output (RO) signal turns on.

Procedure 10-6 Setting Reference Position

- | | |
|-------------|--|
| Step | <ol style="list-style-type: none"> 1 Create a program. 2 Jog the robot to the position that you want to use as a reference position. 3 Record the reference position as a joint position in your program. Do not jog the robot away from this position. 4 Press MENUS. 5 Press STATUS. 6 Select POSITION. 7 Press F2, JNT, if the joint position information is not already displayed. 8 Write down each joint axis position of the taught program position. 9 Press MENUS. 10 Press SETUP. 11 Press F1, [TYPE]. |
|-------------|--|

12 Select Ref Position. You will see a screen similar to the following.

SETUP Ref Pos			JOINT	50 %
			GROUP: 1	1/3
No.	Enb/Dsbl	@Pos	Comment	
1	DISABLE	FALSE	[]
2	DISABLE	FALSE	[]
3	DISABLE	FALSE	[]
[TYPE] GRP#			DETAIL	ENABLE DISABLE

13 To set up the reference position, press F3, DETAIL. You will see a screen similar to the following.

REF POSN			JOINT	50 %
Reference Position			GROUP: 1	1/12
Ref. Position Number:			1	
1	Comment:		[safe position]
2	Enable/Disable:		ENABLED	
3	Signal definition:		DO [1]	
4	J1 :	0.000	+/-	0.000
5	J2 :	0.000	+/-	0.000
6	J3 :	. 0.000	+/-	0.000
7	J4 :	.0.000	+/-	0.000
8	J5 :	.0.000	+/-	0.000
9	J6 :	0.000	+/-	0.000
[TYPE]				RECORD

14 To change the group number,

Enter Group No.: 1

a Press SHIFT and F2.

b Type the group number you want to use for the reference position.

15 Set up the reference position.

a **To change the reference position number**, move the cursor to the number and enter a new value.

b **To enter a comment**, move the cursor to comment and press ENTER. Use the alphanumeric entry as you normally would.

c **To define the signal**, choose either an RO for robot output or DO for digital output and enter the signal number.

d **To enter the joint axis locations**, using the information you wrote down from step 8 to enter the position information.

e **To enter a tolerance range**, move the cursor to the tolerance range and enter the tolerance range.

16 Press PREV to return to the previous screen.

10.7 POSITION REGISTER LOOK-AHEAD EXECUTION FUNCTION

While the robot is executing a program, it reads the lines ahead of the line currently being executed (look-ahead execution). The position register look-ahead execution function enables look-ahead execution for position registers. To understand fully the features of the position register look-ahead function, it is helpful to understand some of the details of program execution.

Program Execution without the Position Register Look-Ahead Function

Conventionally, look-ahead execution is performed for motion instructions that have normal positional data (do not use position registers). Look-ahead execution can not be performed for motion instructions that use position registers for their positional data.

Motion instructions that use position registers can not be read in advance because the values in the position registers could be changed by the program, data transfer function, and so forth.

If the robot reads a motion instruction that uses a position register prior to its execution, the value of the position register might yet be changed by a program or another function (such as data transfer). Such a change is not reflected in the motion instruction that has already been read by the robot. Consequently, the robot's operation might be unpredictable.

Motion instructions that use position registers can be classified into two types:

- Motion instructions with the target position specified by a position register
- Motion instructions with an offset instruction where an offset is given by a position register

Even when a target position or offset is calculated during program execution, and a position register holding this calculation result is used with a motion instruction, look-ahead execution is not performed for the instruction, for the reason explained above.

Program Execution with the Position Register Look-Ahead Function

The position register look-ahead execution function enables look-ahead execution for position registers. For this purpose, an instruction to lock position registers and an instruction to unlock position registers are provided. Using these instructions, you can explicitly specify a program portion. Then, for the specified program portion, even when it contains motion instructions that use position registers, look-ahead execution can be performed.

The position registers can be locked to prevent their contents from being changed after they are read. When an attempt is made to execute an instruction to change a locked position register (for example, an assign instruction for the position register, or an application instruction to set data in the position register), the following error message is issued:

INTP-128 Pos reg is locked

When a function other than the program (such as the data transfer function) attempts to change the value of a locked position register, the following error message is issued, and the attempt fails:

VAR5-053 Pos reg is locked

Position registers are generally locked and unlocked with instructions taught in a program. When a program that has locked the position registers terminates, the position registers are unlocked automatically.

All position registers are locked simultaneously. While the position registers are locked, access to any position register is disabled, even in a different motion group.

10.7.1 Program Instructions

The following program instructions have been added for the position register look-ahead function:

- LOCK PREG
- UNLOCK PREG

LOCK PREG

Locks all position registers. This instruction prevents any change being made to any position register.

UNLOCK PREG

Unlocks the position registers.

These are control instructions, not motion instructions. They can be taught in the same way as other control instructions.

10.7.2 Program Example

Figure 10-26 shows how to use the LOCK PREG and UNLOCK PREG instructions in a program.

Figure 10-26. Position Register Look-Ahead Program Example

```
1: J P[1] 100% FINE
2:   PR[1]=PR[2]
3:   PR[2]=PR[3]
4: LOCK PREG
5: L P[2] 100mm/sec Cnt100
6: L P[3] 100mm/sec Cnt100
7: L PR[1] 100mm/sec Cnt100
8: L P[4] 100mm/sec Cnt100 offset, PR[2]
9: L P[5] 100mm/sec FINE
10: UNLOCK PREG
```

When line 4 of this sample program has been executed, the position registers are locked. They are unlocked when line 10 has been executed. Therefore, the motion instructions with position registers in lines 7 and 8, which are executed with the position registers locked, are subject to look-ahead execution.

If the program is terminated between lines 4 and 10, the locked position registers are unlocked automatically.

If the program is paused between lines 4 and 10, the cursor is moved manually, then the program is restarted, the locked position registers are unlocked. In this case, look-ahead execution is not performed for the instructions in lines 7 and 8.

When backward execution is performed, then normal execution is restarted, the position registers are unlocked. For example, suppose that program execution is paused during the execution of line 6, backward program execution is performed up to line 5, then forward program execution is restarted. In this case, the position registers are unlocked. So, look-ahead execution is not performed for lines 7 and 8.

When program execution is started from a line located after line 4, the position registers are not locked. So, look-ahead execution is not performed for lines 7 and 8.

A LOCK PREG instruction can be executed even when the position registers are already locked. Nothing occurs, however, when the LOCK PREG instruction is executed for a second time. Similarly, the UNLOCK PREG instruction can be executed even when the position registers are not locked. Nothing occurs, however, when the UNLOCK PREG instruction is executed for a second time.

10.7.3 Execution

When executing position register look-ahead program instructions, be aware of the following:

- The LOCK PREG and UNLOCK PREG instructions are not executed in backward program execution mode.
- Look-ahead execution is not performed for the LOCK PREG and UNLOCK PREG instructions. This means that when one of these instructions is encountered, look-ahead execution is stopped temporarily; after the instruction is executed, look-ahead execution is again enabled.

10.8 SHAPE GENERATION (OPTION)

The shape generation option simplifies cutting two dimensional shapes. This shape generation software reduces the on-line programming time by reducing the number of required taught robot positions and providing features like shape cut macros and on-the-fly shape shift.

You can generate the following shapes using the shape generation option:

- Circles
- Hexagons
- Rectangles
- Slots

To use the shape generation option, you must do the following:

1. Set up shape information – Section 10.8.1
2. Set up shape schedules for the kinds of shapes you want to generate – Section 10.8.2
3. Include the instructions within a program to perform shape generation – Section 10.8.3

NOTE When you install Shape Generation Software, be sure that HandlingTool V4.3 with AccuPath and Small Circle Accuracy Options has been loaded on the RJ-2 controller. AccuPath and Small Circle Accuracy are supported on certain robot models only. To load the Shape Generation Software option, refer to the *application-specific SYSTEM R-J2 Software Installation Manual*.



WARNING

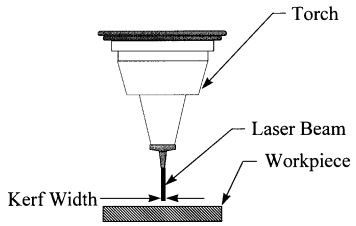
Make sure your program stops the process (such as laser or water jet cutting) when an error occurs. The Shape Generation option does not automatically turn the process equipment off when an error occurs. For example, if you are cutting shapes using a laser or water jet, you must write the program so that the process shuts off if an error occurs. If you do not, you could injure personnel or damage equipment.

10.8.1 Shape Setup

You must set up specific information before you can use the shape generation option. You specify this information in the SETUP Shapes screen. The information you specify in this screen will apply to all shapes you generate using the shape generation option.

Table 10-5 lists and describes the items in the SETUP Shapes screen. Use Procedure 10-7 to set up shape information.

Table 10-5. Shape Setup Information

ITEM	DESCRIPTION
Kerf width	<p>The Kerf Width defines the corrective value for the material taken away during the cutting process. This distance is specified in mm between 0.0 mm and 5.0 mm. The following diagram further illustrates the kerf width.</p>  <p>The diagram shows a cross-section of a laser cutting process. A torch is positioned above a workpiece. A laser beam is directed from the torch onto the workpiece, creating a cut. The width of the cut is labeled as 'Kerf Width'. Labels include 'Torch', 'Laser Beam', 'Workpiece', and 'Kerf Width'.</p>
Cut direction	<p>Defines the direction of motion for all shapes. You can specify a Clockwise or Counter clockwise cut direction by pressing F4 or F5 when the cursor is on the cut direction field.</p>
Shape Schedule Register	<p>Defines the Register number for specifying the schedule of a given Shape type. This can be set from 1 to 10, 32, or 64 (depending on how many registers your system has). The default value is R[1].</p>
Center Position Register	<p>Defines the Position Register number for specifying the center of a given Shape type. This can be set from 1 to 10, 32, or 64 (depending on how many position registers your system has). The default value is PR[1].</p>
Rotation Register	<p>Defines the Register Number assigned to implement a shape rotation. The value of the Rotation Register is in degrees and can be a positive or negative value from 0 to 360. The default value is R[2]. (Refer to Section 10.8.5, "Shape Adjust Utility").</p>
Kerf Width Override	<p>Defines the Register Number assigned to override the current kerf width. The default value is R[3]. (Refer to Section 10.8.3, "Programming").</p>
Cut Direction Override	<p>Defines the Register Number assigned to override the current cut direction. The default value is R[4]. (Refer to Section 10.8.3, "Programming").</p>
Speed Override	<p>Defines the Register Number assigned to override the current speed specified by the Shape Schedule. The default value is R[5]. (Refer to Section 10.8.3, "Programming").</p>
Step Through Center	<p>Defines whether the robot, when in Step Mode, steps to the shape center (Enabled) or steps directly to the pierce position (Disabled).</p>

Procedure 10-7 Setting Up Shape Information

- Step**
- 1 Press **MENUS**.
 - 2 Select **SETUP**.
 - 3 Press **F1**, **[TYPE]**.
 - 4 Select **Shapes**. You will see a screen similar to the following.

SETUP Shapes		JOINT 10 %
		1/9
1 Kerf width:	0.00 mm	
2 Cut direction:	CW	
3 Shape schedule register:	R[1]	
4 Center position register:	PR[1]	
5 Rotation register:	R[2]	
6 Kerf width override:	R[3]	
7 Cut direction override:	R[4]	
8 Speed override:	R[5]	
9 Step through Center:	ENABLE	
[TYPE]		HELP

- 5 Move the cursor to each item and set as desired. Refer to Table 10-5.

10.8.2 Shape Schedules

You can set up schedules to define how you want to generate the following shapes:

- Circle
- Hexagon
- Rectangle
- Slot

This section contains

- A procedure for using shape SCHEDULE and DETAIL screens
- Schedule screens contain information for circle, hexagon, rectangle, and slot shapes

Using the SCHEDULE and DETAIL Screens

While the schedule screen is being displayed a table summary of all schedules is displayed. Procedure 10-8 describes how to use the Shape SCHEDULE and DETAIL screen. The circle SCHEDULE and DETAIL screens are shown as an example.

Procedure 10-8 Using the Shape SCHEDULE and DETAIL Screen

- Step**
- 1 Press DATA.
 - 2 Press F1, [TYPE].
 - 3 Select the kind of shape you want (Circle, Hex, Rect or Slot). You will see a screen similar to the following.

DATA Circles				JOINT 10 %	
					1/100
#	Dia.	Cm/min	Type	[Comment
1	55.00	555	FULL	CIRCLE	1
2	12.00	123	HALF	12 mil	half
3	25.00	456	FULL	25 mil	circ
4	0.00	380	FULL		
5	0.00	380	FULL		
6	0.00	380	FULL		
7	0.00	380	FULL		
8	0.00	380	FULL		
[TYPE] DETAIL UNITS					HELP >

- 4 To display the **DETAIL** screen, press F2, **DETAIL**. You will see a screen similar to the following.

DATA Circles		JOINT 10 %
		1/10
1	Schedule #: 1	[CIRCLE 1]
2	Diameter:	55.00 mm
3	Speed:	380 cm/min
4	Circle Type:	FULL
5	Start Axis:	+X
6	X Work angle:	0.00 deg
7	Y Work angle:	0.00 deg
8	Blend-in angle:	20.00 deg
9	Blend-in distance:	3.00 mm
10	Overlap angle:	10.00 deg
[TYPE] SCHEDULE UNITS		HELP >

```
Enter schedule number to display:
[ TYPE ] SCHEDULE      HELP
```

```
Enter schedule number to copy to:
[ TYPE ] SCHEDULE      HELP
```

```
Clear current schedule?
                YES  NO
```

- 5 To select a new schedule, press F2, **SCHEDULE**.
- 6 To copy the current schedule, press **NEXT, >**, and then press F2, **COPY**. It will prompt you to type a new schedule number to which the current schedule will be copied.
- 7 To reset the current schedule to default values, press **NEXT, >**, then press F3, **CLEAR**.
- 8 To display help information, press F5, **HELP**.
- 9 To display the previous screen, press **PREV**.
- 10 To toggle the speed units, press F3, **UNITS**. The speed units will change between mm/sec to cm/min to in/min.

Circle Schedules

Up to 100 full and half circle schedules are available. Figure 10-27 shows the circle schedule screen and Figure 10-28 shows the circle schedule DETAIL screen. The items on these screens are listed and described in Table 10-6.

Figure 10-27. Circle Schedule SCHEDULE Screen

DATA Circles				JOINT 10 %
				1/100
#	Dia.	Cm/min	Type	[Comment]
1	55.00	555	FULL	CIRCLE 1
2	12.00	123	HALF	12 mil half
3	25.00	456	FULL	25 mil circ
4	0.00	380	FULL	
5	0.00	380	FULL	
6	0.00	380	FULL	
7	0.00	380	FULL	
[TYPE] DETAIL UNITS				HELP >

Figure 10-28. Circle Schedule DETAIL Screen

DATA Circles		JOINT 10 %
		1/10
1	Schedule #: 1	[CIRCLE 1]
2	Diameter:	55.00 mm
3	Speed:	380 cm/min
4	Circle Type:	FULL
5	Start Axis:	+X
6	X Work angle:	0.00 deg
7	Y Work angle:	0.00 deg
8	Blend-in angle:	20.00 deg
9	Blend-in distance:	3.00 mm
10	Overlap angle:	10.00 deg
[TYPE]SCHEDULE UNITS		HELP >

Table 10-6. Circle Schedule Data

ITEM	DESCRIPTION
Schedule # Range: (1 - 100)	The current shape schedule number.
Comment	Each schedule can be commented for ease of use.
Diameter Range (1 - 500 mm)	The diameter in mm of the current circle/half circle.
Speed Range: (1 - 9999 cm/sec)	The cutting speed of the circle/half circle.
Circle Type (Full/Half Circle)	Specifies if circle is full or half circle.

Table 10-6. (Cont'd) Circle Schedule Data

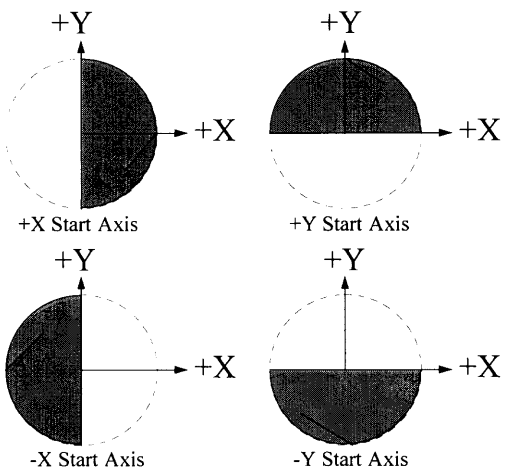
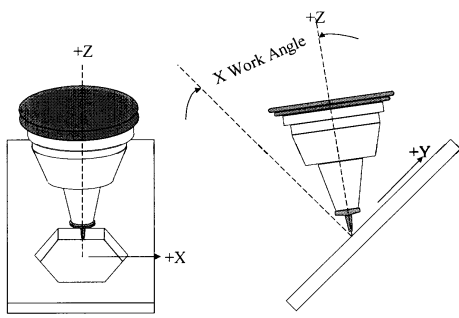
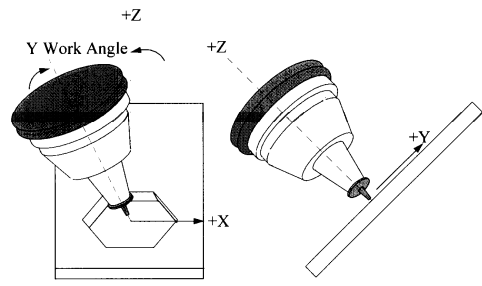
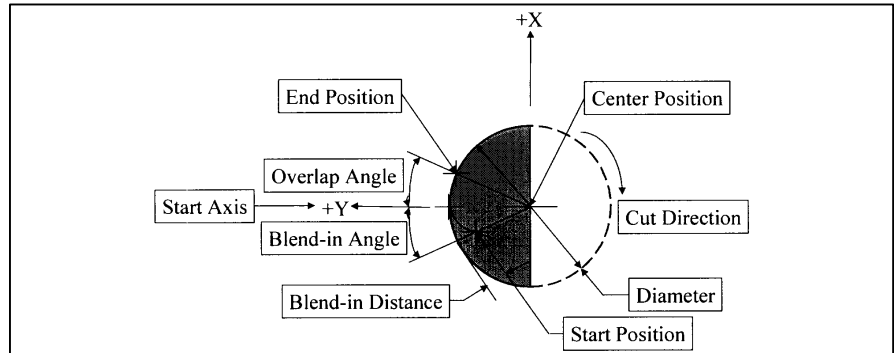
ITEM	DESCRIPTION
<p>Start Axis (+/-X or +/-Y)</p>	<p>The start axis is defined as which axis on the shape that the robot will begin its cut. Shown below are each of the four (4) start axis choices.</p> <p>Note: Half circle orientation depends on which start axis is implemented.</p> 
<p>X-Work Angle Range: (+/- 45°)</p>	<p>The X and Y Work angles are provided to allow you to program beveled cuts. The angles are measured from the normal around the corresponding shape axis (X or Y).</p> 
<p>Y-Work Angle Range: (+/- 45°)</p>	
<p>Blend-In Angle Range: (0 – 45°)</p>	<p>The angle from the start axis to the start position. See Figure 10-29.</p>
<p>Blend-In Distance Range: (0 – 20 mm)</p>	<p>The distance from the shape edge to start position. See Figure 10-29.</p>
<p>Overlap Angle Range: (0 – 45°)</p>	<p>The angle between the start axis and end position. See Figure 10-29.</p>

Figure 10-29. Circle Schedule Terminology



Hexagon Schedules

Up to 50 hexagon schedules are available. Figure 10-30 shows the hexagon schedule screen and Figure 10-31 shows the hexagon schedule DETAIL screen. The items on these screens are listed and described in Table 10-7.

Figure 10-30. Hexagon Schedule SCHEDULE Screen

DATA Hexagon				JOINT 10 %
				1/50
Dia.	Cnr.	Cm/min		[COMMENT]
1	15.00	2.00	350	Hexagon 1
2	25.00	1.00	345	Hexagon 12
3	35.00	1.00	350	1/4 in Hex
4	0.00	0.00	380	
5	0.00	0.00	380	
6	0.00	0.00	380	
7	0.00	0.00	380	
8	0.00	0.00	380	
[TYPE] DETAIL UNITS				HELP >

Figure 10-31. Hexagon Schedule DETAIL Screen

DATA Hexagon		JOINT 10 %
		1/10
1	Schedule #: 1	[Hexagon 1]
2	Diameter:	15.00 mm
3	Corner Radii:	2.00 mm
4	Speed:	350 mm/sec
5	Start Axis:	+X
6	X Work Angle:	5.0 deg
7	Y Work Angle:	15.0 deg
8	Blend-in angle:	20.0 deg
9	Blend-in distance:	5.0 mm
10	Overlap angle:	20.0 deg
[TYPE] SCHEDULE UNITS		HELP >

Table 10-7. Hexagon Schedule Data

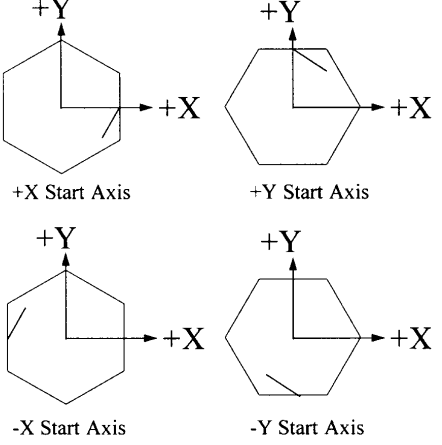
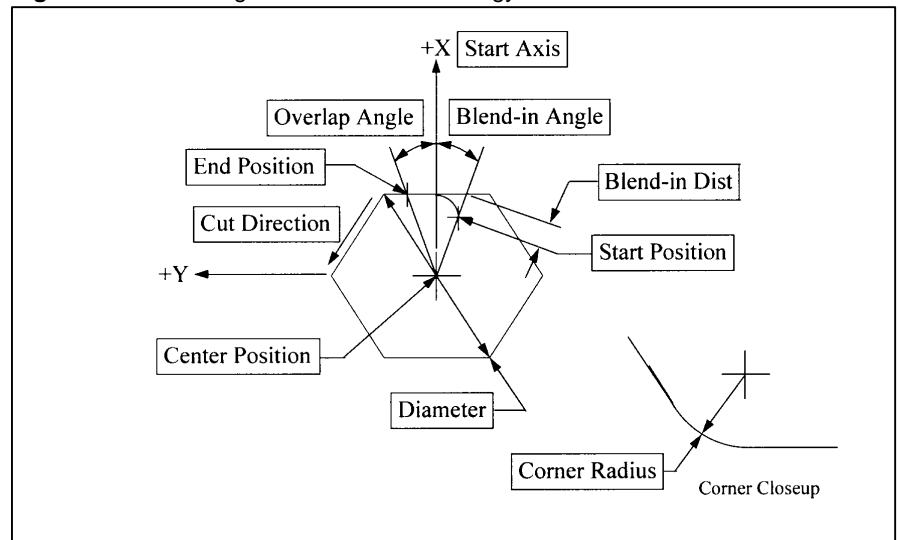
ITEM	DESCRIPTION
Schedule # Range: (1 – 50)	The current schedule to which the defined data corresponds.
Comment	Each schedule can be commented for ease of use.
Diameter Range: (1 – 500 mm)	The diameter in mm across the corners of the current hexagon.
Corner Radii	Specifies the radius of the corners (all corners are identical).
Speed Range: (1 - 9999 cm/sec)	The cutting speed of the current hexagon.
Start Axis Range: (+/- X or +/- Y)	<p>The start axis is defined as which axis on the shape that the robot will begin its cut. Shown below are each of the four (4) start axis choices.</p> <p>Note: The hexagon orientation depends on start axis, cut always starts on a flat edge.</p>  <p>The diagrams illustrate four start axis choices for a hexagon: <ul style="list-style-type: none"> +X Start Axis: The hexagon is oriented with a flat edge on the right side, and the cut starts on this edge. +Y Start Axis: The hexagon is oriented with a flat edge on the top side, and the cut starts on this edge. -X Start Axis: The hexagon is oriented with a flat edge on the left side, and the cut starts on this edge. -Y Start Axis: The hexagon is oriented with a flat edge on the bottom side, and the cut starts on this edge. </p>
X-Work Angle Range: (+/- 45°)	Refer to Table 10-6.
Y-Work Angle Range: (+/- 45°)	
Blend-In Angle Range: (0 – 45°)	The angle from the start axis to the start position. See Figure 10-32.
Blend-In Distance Range: (0 – 20 mm)	The distance from the shape edge to start position. See Figure 10-32.
Overlap Angle Range: (0 – 45°)	The angle between the start axis and end position. See Figure 10-32.

Figure 10-32. Hexagon Schedule Terminology



Rectangle Schedules

Up to 50 rectangle schedules are available. Figure 10-33 shows the rectangle schedule screen and Figure 10-34 shows the rectangle schedule DETAIL screen. The items on these screens are listed and described in Table 10-8.

Figure 10-33. Rectangle Schedule SCHEDULE Screen

DATA Rectangle				JOINT 10 %
				1/50
	Len.	Wid.	Cm/min [COMMENT]
1	15.00	10.00	350	Rect 1
2	20.00	14.50	325	3/4 in. Rect
3	12.50	35.00	355	1/2 in. Rect
4	8.00	6.00	380	8 mm Rect
5	12.00	55.0	380	1/8 Rect
6	0.00	0.0	380	
7	0.00	0.0	380	
8	0.00	0.0	380	
9	0.00	0.0	380	
[TYPE]				DETAIL UNITS HELP >

Figure 10-34. Rectangle Schedule DETAIL Screen

DATA Rectangle		JOINT 10 %
		1/14
1	Schedule #: 1	[Rect 1]
2	Length:	25.00 mm
3	Width:	15.00 mm
4	Speed:	350 cm/min
5	Radius 1:	5.00 mm
6	Radius 2:	2.00 mm
7	Radius 3:	1.00 mm
8	Radius 4:	5.00 mm
9	Start Axis:	+X
10	X Work Angle:	5.0 deg
11	Y Work Angle:	10.0 deg
12	Blend-in angle:	25.0 deg
13	Blend-in distance:	5.0 mm
14	Overlap angle:	25.0 deg
[TYPE]		SCHEDULE UNITS HELP >

Table 10-8. Rectangle Schedule Data

ITEM	DESCRIPTION
Schedule # Range: (1 - 50)	The current schedule to which the defined data corresponds.
Comment	Each schedule can be commented for ease of use.
Length Range: (1 - 500 mm)	The length in mm across the longer side of the current rectangle.
Width Range: (1 - 500 mm)	The width in mm across the shorter side of the current rectangle.

Table 10-8. (Cont'd) Rectangle Schedule Data

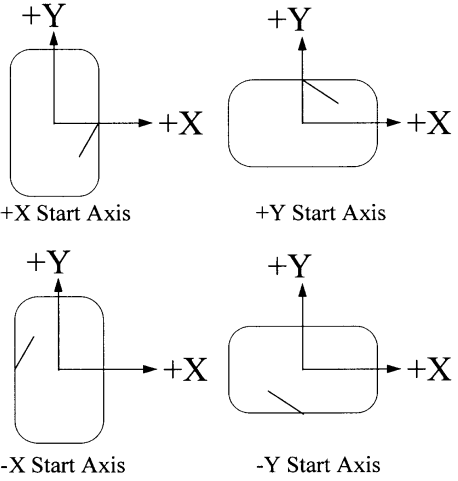
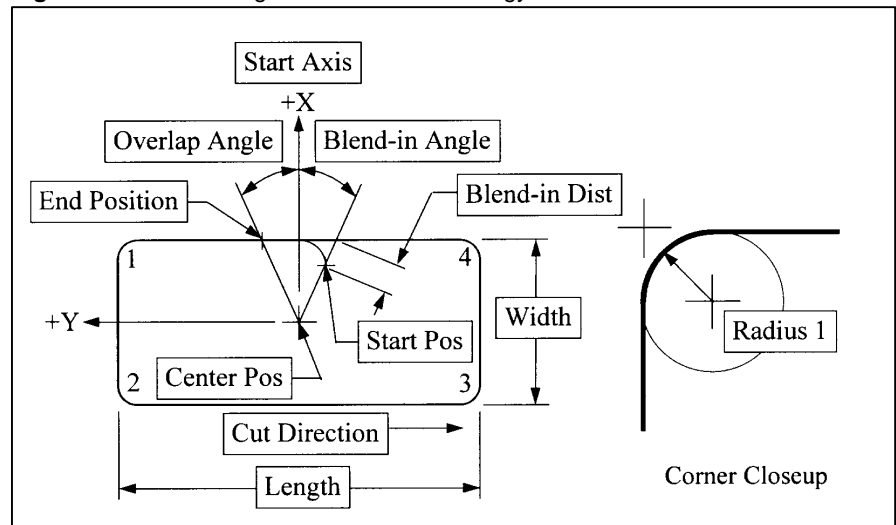
ITEM	DESCRIPTION
Speed Range: (1 - 9999 cm/sec)	The cutting speed of the current rectangle.
Radius 1	Specifies the radius of the first corner encountered in the rectangle depending on cut direction.
Radius 2	Specifies the radius of corner 2.
Radius 3	Specifies the radius of corner 3.
Radius 4	Specifies the radius of corner 4.
Start Axis Range: (+/-X or +/- Y)	<p>The start axis is defined as which axis on the shape that the robot will begin its cut.</p> <p>NOTE: The rectangle orientation depends on start axis is implemented. The cut always begins on the center of the longest side.</p> 
X-Work Angle Range: (+/- 45°)	Refer to Table 10-6.
Y-Work Angle Range: (+/- 45°)	
Blend-In Angle Range: (0 - 45°)	The angle from the start axis to the start position. See Figure 10-35.
Blend-In Distance Range: (0 - 20 mm)	The distance from the shape edge to start position. See Figure 10-35.
Overlap Angle Range: (0 - 45°)	The angle between the start axis and end position. See Figure 10-35.

Figure 10-35. Rectangle Schedule Terminology



Slot Schedules

Up to 50 slot schedules are available. Figure 10-36 shows the rectangle schedule screen and Figure 10-37 shows the rectangle schedule DETAIL screen. The items on these screens are listed and described in Table 10-9.

Figure 10-36. Slot Schedule SCHEDULE Screen

DATA Slot				JOINT 10 %
Len.	Wid.	Cm/min	[COMMENT]
1	15.00	10.00	350	Slot 1
2	20.00	14.50	225	3/4 in. Slot
3	12.50	35.00	255	1/2 in. Slot
4	8.00	6.00	220	8 mm Slot
5	12.00	55.0	200	1/8 Slot
6	0.00	0.0	250	
7	0.00	0.0	250	
8	0.00	0.0	250	
9	0.00	0.0	250	
[TYPE]				DETAIL UNITS HELP >

Figure 10-37. Slot Schedule DETAIL Screen

DATA Slot	JOINT 10 %
1 Schedule #: 1	[Slot 1 1/10]
2 Length:	15.00 mm
3 Width:	10.00 mm
4 Speed:	350 cm/min
5 Start Axis:	+X
6 X Work Angle:	5.0 deg
7 Y Work Angle:	10.0 deg
8 Blend-in angle:	25.0 deg
9 Blend-in distance:	5.0 mm
10 Overlap angle:	25.0 deg
[TYPE]SCHEDULE UNITS	HELP >

Table 10-9. Slot Schedule Data

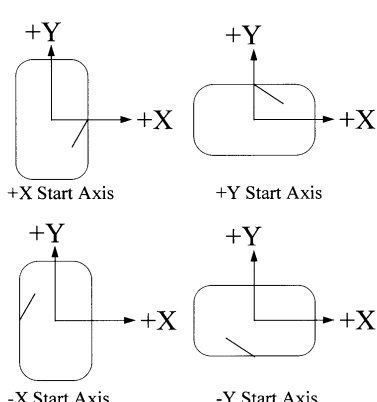
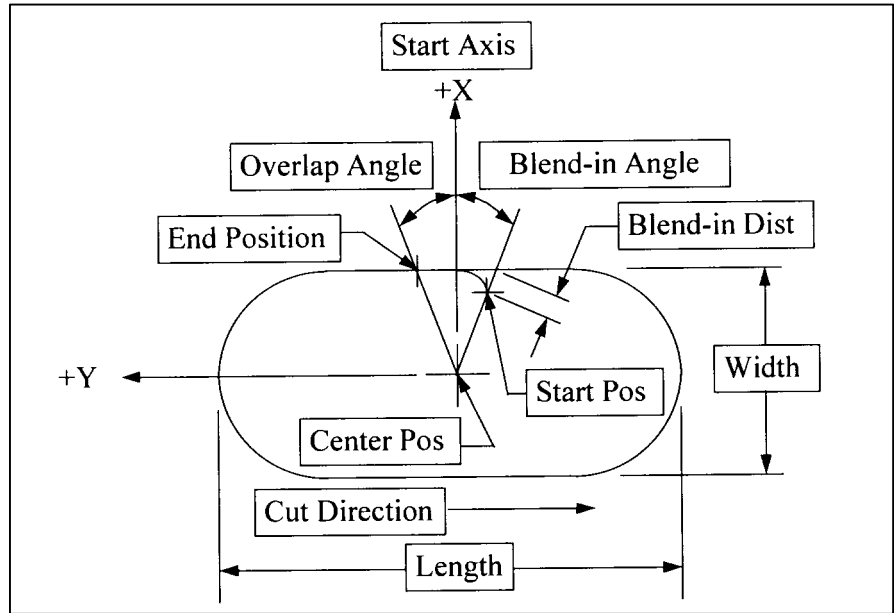
ITEM	DESCRIPTION
Schedule # Range: (1 - 50)	The current schedule to which the defined data corresponds.
Comment	Each schedule can be commented for ease of use.
Length Range: (1 - 500 mm)	The length in mm across the longer side of the current slot.
Width Range: (1 - 500 mm)	The width in mm across the shorter side of the current slot.
Speed Range: (1 - 9999 cm/sec)	The cutting speed of the current slot.
Start Axis Range: (+/-X or +/- Y)	<p>The start axis is defined as which axis on the shape that the robot will begin its cut.</p> <p>NOTE The rectangle orientation depends on whether the start axis is implemented.</p> 

Table 10-9. (Cont'd) Slot Schedule Data

ITEM	DESCRIPTION
X-Work Angle Range: (+/- 45°)	Refer to Table 10-6.
Y-Work Angle Range: (+/- 45°)	
Blend-In Angle Range: (0 - 45°)	The angle from the start axis to the start position. See Figure 10-38.
Blend-In Distance Range: (0 - 20 mm)	The distance from the shape edge to start position. See Figure 10-38.
Overlap Angle Range: (0 - 45°)	The angle between the start axis and end position. See Figure 10-38.

Figure 10-38. Slot Schedule Terminology



10.8.3 Programming

You must include information in your program to perform shape generation. Instructions and information you include in a program is used by the shape generation feature to generate the shapes you specify.

You must include the following information in your program to perform shape generation:

- A **shape schedule instruction**, in which you assign the shape schedule number you want to use to the **shape schedule register** you assign in the SETUP Shapes screen (Section 10.8.1)
- A **shape center position register instruction**, in which you assign the shape center position register to the position number of the center position.
- A **shape center position instruction** that you must record.
- A **shape calculation macro instruction** to calculate the shape information the robot will use to make the shape.
- A **shape cutting macro instruction** to make the shape.

Figure 10-39 contains an example shape generation program. Table 10-10 summarizes each program instruction.

Figure 10-39. Shape Generation Program Example

```

JOB123                                JOINT 10 %
                                         6/6

1: R[1:Shape Schedule]=1
2: PR[1: Center ]=P[2: Hole 1 ]
3: Calc Circle
4:J P[2: Hole 1 ] 100% FINE
   : Offset,PR[2: Start Offset]
5: Cut Circle

POINT                                TOUCHUP>

```



WARNING

In your shape generation program, you must include the program instructions in the order shown in this example for the proper execution of the shape generation option. Otherwise, you could injure personnel or damage equipment.

**WARNING**

Make sure your program stops the process (such as laser or water jet cutting) when an error occurs. The Shape Generation option does not automatically turn the process equipment off when an error occurs. For example, if you are cutting shapes using a laser or water jet, you must write the program so that the process shuts off if an error occurs. You can use the process on and process off macros for this purpose (refer to Table 10-12). If you do not, you could injure personnel or damage equipment.

NOTE You cannot use the following registers, positions registers, and UFRAME numbers in your shape generation programs:

- R[7] – used for the full/half circle flag
- PR[2] – used for the center position offset
- UFRAME[1,5] – used for shape frames

NOTE Use descriptive comments for all registers, position registers, and positions for easy identification during programming, testing, and production.

Table 10-10. Shape Generation Program Example Description

Line Number	Description
1	The shape schedule register (set in the SETUP Shapes screen) stores the desired shape schedule number, which is 1 in this example.
2	The shape center position register (set in the SETUP Shapes screen) stores the positional data of the taught center position.
3	The Calc Shape macro instruction (Calc Circle in this example) calculates the shape positions and stores them in the Cut Shape macro.
4	The taught center position, offset by PR[2], is calculated in the Calc Shape macro for teach and production mode operation.
5	The Cut Shape macro instruction contains all positions required to cut the required shape, which were calculated using the Calc Shape macro instruction.

Shape Macros

The shape generation option requires that you use the following macros in your shape generation program:

- Calculate Shape
- Cut Shape

Calculate Shape Macros

The Calculate Shape macros calculate the shape positions and store them in the corresponding Cut Shape macros. One Calculate Shape macro is provided for each kind of shape.

The Calculate Shape macros call KAREL routines that use shape setup information, schedules, and register information to calculate shape positions and store them in the Cut Shape macros. The Calculate Shape macros also determine whether to move the robot to the center position or the pierce point on the shape. Table 10-11 lists the Calculate Shape macros.

Table 10-11. Calculate Shape Macros

Macro Name	Macro Program Name
Calculate Circle	Calc_cir.tp
Calculate Hexagon	Calc_hex.tp
Calculate Rectangle	Calc_rec.tp
Calculate Slot	Calc_slit.tp

Cut Shape Macros

The Cut Shape macros contain all robot positions necessary to cut the required shapes. These Cut Shape macros call macros that can be customized for I/O control. Table 10-12 lists the Cut Shape macros.



WARNING

Do not change any information in the Cut Shape macros. If you modify a Cut Shape macro, you might disrupt normal operation and injure personnel or damage equipment.

Table 10-12. Shape Macros

Macro Name	Macro Program Name
Cut Circle	Cut_cir.tp
Cut Hexagon	Cut_hex.tp
Cut Rectangle	Cut_rec.tp
Cut Slot	Cut_slit.tp
Process On	Proc_on.tp
Process Off	Proc_off.tp

Process Macros

Within the cut shape macros are calls to two process macros:

- Process On
- Process Off

These macros are empty teach pendant programs that enable limit process control by allowing you to program I/O manipulations.

**WARNING**

Process macros must not contain any motion statements. Otherwise, you could injure personnel or damage equipment.

The Process On macro is called at the beginning of each cut shape macro. The Process Off macro is called at the end of each cut shape macro.

Using Override Registers

For added flexibility, you can override the following information in your shape generation program, using an **override register**:

- Kerf width
- Speed
- Cut direction

An override register is a register that contains a value that will change the corresponding value in the shape schedule to the value you specify in the override register. Override registers have the following **characteristics**:

- The information specified in an override registers affects the next Calculate Shape macro only.
- All override register values are reset after each Calculate Shape macro.
- Override registers affect only one shape.
- Setting override registers to zero does not affect schedule information.

Figure 10-40 shows an example of using override registers.

Figure 10-40. Example of Using Override Registers

```

JOB123                                JOINT 10 %
                                         5/8
1: R[1:Shape Schedule]=5
2: R[3:Kerf Wid Ovr] = 1.25
3: R[4:Cut CW Ovr] = 1
4: R[5:Spd Ovr (mm/s)] = 250
5: PR[1: Center ]=P[2: Hole 1 ]
6: Calculate Circle
7: J P[2: Hole 1 ] 100% FINE
   : Offset,PR[2: Start Offset]
8: Cut Circle

POINT                                TOUCHUP>

```

In this program, 2, 3 and 4 override the information in circle schedule 5. The Calculate Circle macro will then use the override information instead of the schedule information. The overrides used are as follows:

- The Kerf Width override has a range of 0.000 to 4.999 mm.
- The Speed override register has a range of 0 to 999 cm/min.
- Any non-zero value set in the Cut Direction override register changes the current cut direction (set in the SETUP Shapes screen) from CW to CCW or from CCW to CW.

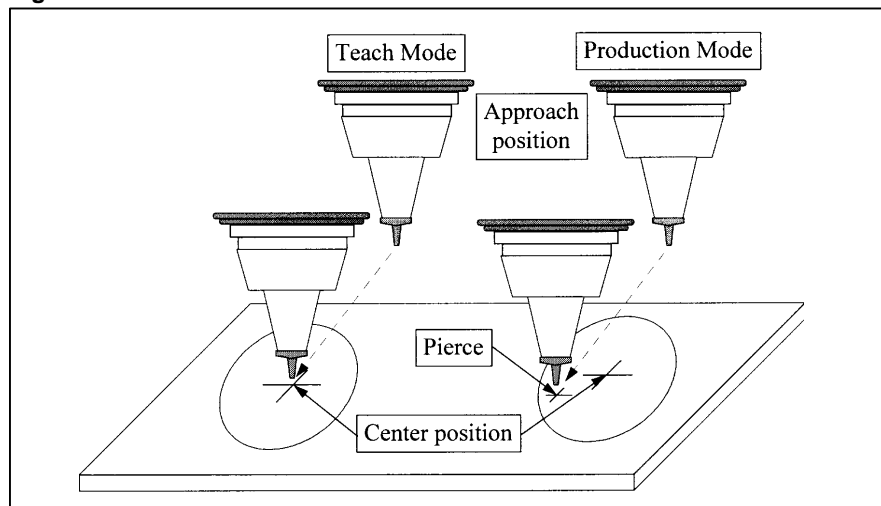
10.8.4 Teach and Production Modes

The shape generation option has the following modes of operation:

- **Teach mode** – While the robot is in step mode, teach mode is enabled. If Step through Center (set up on the SETUP:Shape screen) is enabled, the robot will step to the center position. If Step through Center is disabled, the robot will step to the pierce positions.
- **Production mode** – The robot will use the center position offset to move to the pierce position of the shape. While the robot is in production mode, step mode is disabled.

See Figure 10-41.

Figure 10-41. Teach and Production Modes



10.8.5 Shape Adjust Utility

The Shape Adjust utility allows you to shift programmed shapes in two dimensions. The shapes can be offset in X or Y, along a frame relative to their center position (shape frame). You cannot offset the shapes relative to WORLD frame or the taught user frame.

You use the Shape Adjust Utility using the UTILITIES Shape Adjust screen. See Figure 10-42 and Figure 10-43 for the UTILITIES Shape Adjust SCHEDULE and DETAIL screens.

Figure 10-42. UTILITIES Shape Adjust SCHEDULE Screen

UTILITIES Shape Adjust			JOINT 10 %
Program	Line #	Status	1/10
1 CIRCLE	6	ENABLED	
2 JOB123	19	EDIT	
3 RR_RAIL	27	DISABLED	
4 *****	0	*****	
5 *****	0	*****	
6 *****	0	*****	
7 *****	0	*****	
8 *****	0	*****	
9 *****	0	*****	
10 *****	0	*****	
[TYPE] DETAIL UNITS			>

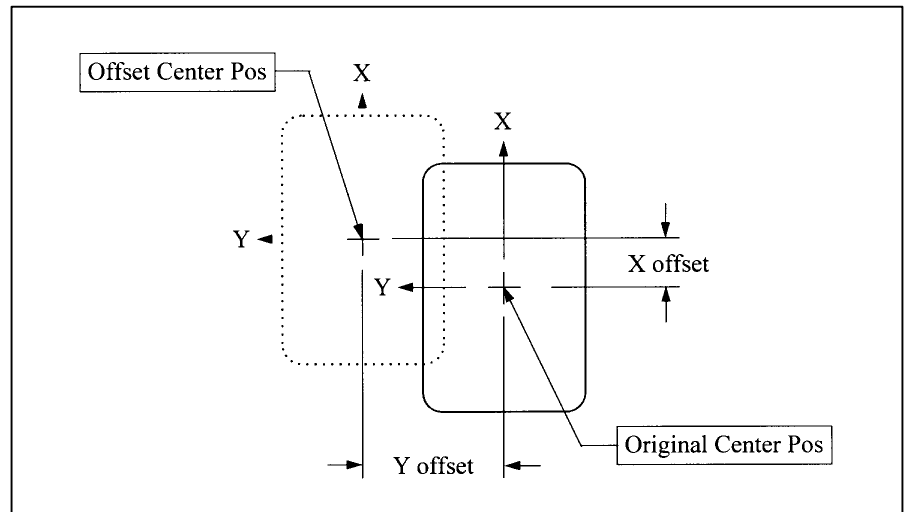
Figure 10-43. UTILITIES Shape Adjust DETAIL Screen

UTILITIES Shape Adjust			JOINT 10 %
			1/4
Current schedule: 1 Status: ENABLED			
1 Program name:	CIRCLE		
2 Line number:	6		
3 X offset:	10.000	mm	
4 Y offset:	5.000	mm	
[TYPE]			SCHED DISABLE >

X and Y Offsets

X and Y offsets entered into the Shape Adjust screen will adjust a shape's position the next time you run the program. This new offset shape position will be used in the program until you disable or change the adjustment in the UTILITIES Shape Adjust screen. **The shape will be shifted along its shape frame. The shape will not be adjusted along the WORLD frame or the taught user frame.** See Figure 10-44.

Figure 10-44. Shape Adjust Utility: X and Y Offsets



Shape Rotation

Shape rotation is part of the shape offset capability. The SETUP Shapes screen allows you to identify a rotation register. When this register is assigned a value in your program, the corresponding shape will be rotated around its Z axis by the specified amount. See Figure 10-45.

Figure 10-45. UTILITIES Shape Adjust: Shape Rotation

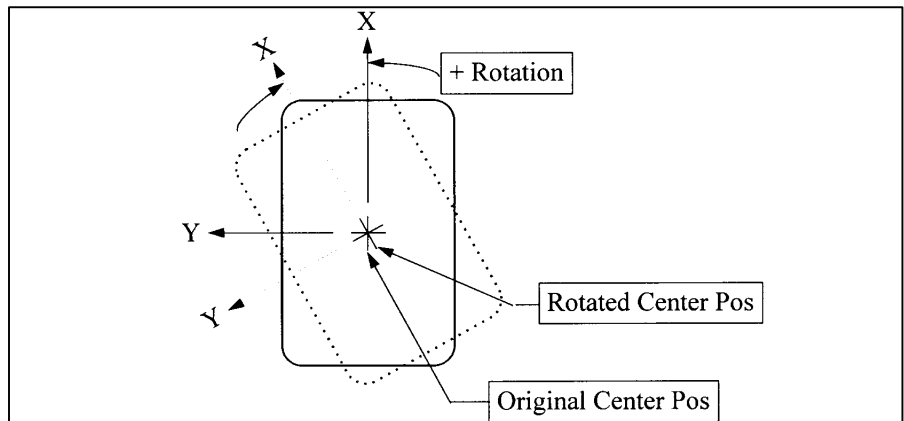


Figure 10-46 contains a typical programming example with a shape rotation. Program line 3 specifies the rectangle be rotated 30° .

Figure 10-46. Shape Rotation Program Example

```

JOB123                               JOIN 10 %
                                         6/6
1: R[20:Shape Schedule]=5
2: PR[1: Center ]=P[2: Hole 1 ]
3: R[2:Rotation]=30
4: Calculate Rectangle
5: J P[2: Hole 1 ] 100% FINE
   : Offset,PR[2: Start Offset]
6: Cut Rectangle

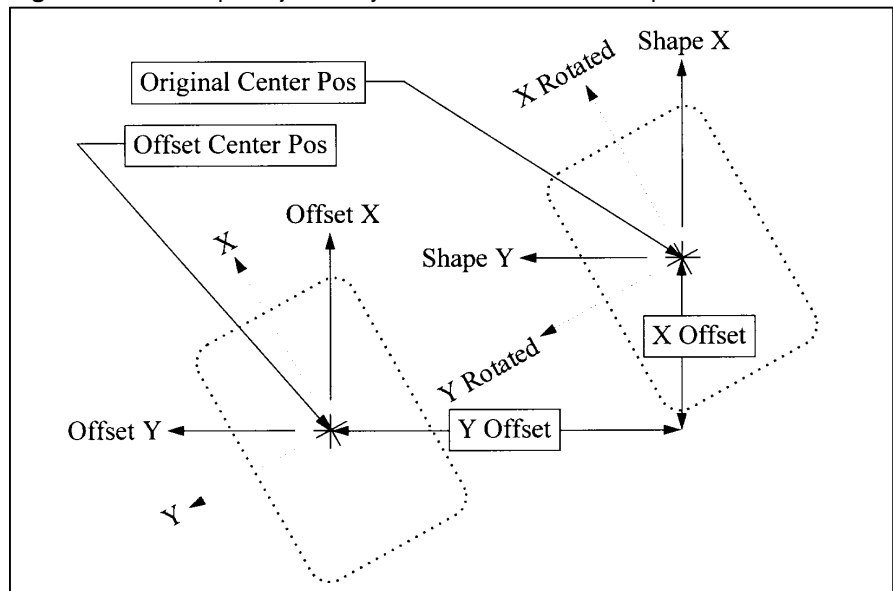
POINT                                TOUCHUP>

```

Offset and Rotated Shapes

Rotated shapes are shifted along the shape frame. They are not shifted along the WORLD frame or the taught user frame. Figure 10-47 shows how an X-Y applies to a rotated shape.

Figure 10-47. Shape Adjust Utility: Offset and Rotated Shapes



Shape Adjust Procedure

Use Procedure 10-9 to perform shape adjustments using the Shape Adjust Utility.

Procedure 10-9 Performing Shape Adjustments Using the Shape Adjust Utility

- Step**
- 1 Press **MENU**.
 - 2 Select **UTILITIES**.
 - 3 Press **F1**, [**TYPE**].
 - 4 Select **Shape Adjust**. You will see a screen similar to the following.

UTILITIES Shape Adjust			JOINT 10 %
Program	Line #	Status	1/10
1 CIRCLE	6	ENABLED	
2 JOB123	19	EDIT	
3 RR_RAIL	27	DISABLED	
4 *****	0	*****	
5 *****	0	*****	
6 *****	0	*****	
7 *****	0	*****	
8 *****	0	*****	
9 *****	0	*****	
10 *****	0	*****	
[TYPE] DETAIL UNITS			>

- 5 To display the **DETAIL** screen, press **F2**, **DETAIL**. You will see a screen similar to the following.

UTILITIES Shape Adjust			JOINT 10 %
			1/4
Current schedule: 1 Status: ENABLED			
1 Program name:	CIRCLE		
2 Line number:	6		
3 X offset:	10.000	mm	
4 Y offset:	5.000	mm	
[TYPE]			SCHED DISABLE >

- 6 Select and set items as desired.

10.8.6 Shape Frames

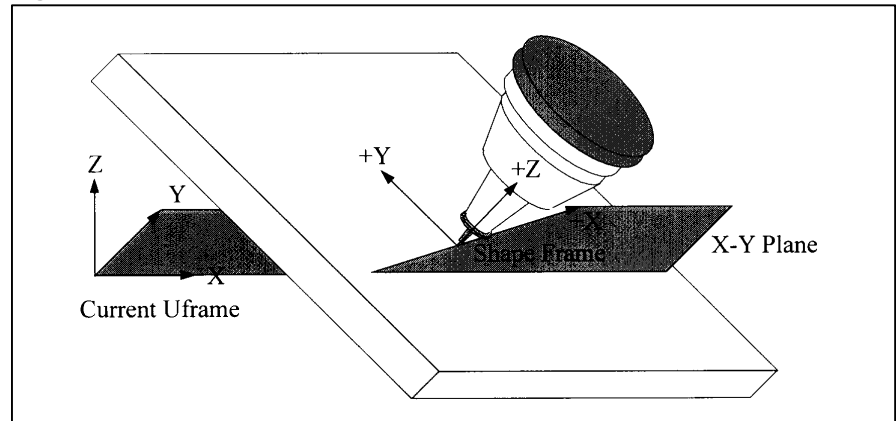
The shape generation software utilizes shape frames. A shape frame is calculated from a shape center position and current uframe.

The shape frame X-direction is parallel to the object surface where the current Uframe X-Y plane intersects the object at the shape center position.

The shape frame Y-direction is also parallel to the object surface but perpendicular to the X-direction.

The shape frame Z-direction is into the tool along the Tool-Z vector. See Figure 10-48.

Figure 10-48. Shape Frames



10.9 MOTION GROUP DO OUTPUT FUNCTION

If the multiple motion group option is used to operate more than one robot connected to one R-J2 controller unit, a program can control the multiple robots simultaneously. However, only one robot can be jogged manually using the teach pendant.

A motion group used by a program is recorded as detail information for the program. The motion group that can be jogged can be switched using the teach pendant.

During teaching, the teach pendant screen indicates the motion group to which a robot that can be jogged belongs. Conventionally, this information can not be known by externally without using the teach pendant.

The motion group DO output function outputs information about motion groups that can be jogged or motion groups a paused or running program uses as a digital output signal (DO) or a robot output signal (RO). In this way, it is possible to recognize the currently enabled motion groups without using the teach pendant, thus enhancing safety.

NOTE The motion group DO output function is available only when the multiple motion group option is used.

10.9.1 Restrictions

Be aware of the following restrictions when using the motion group DO output function:

- The same signal cannot be defined for different motion groups.
- When a program is running or paused, the type of signal (DO/RO) or signal number cannot be changed.

10.9.2 Operations

The motion group DO output function can assign two types of digital output signals to one motion group: jog signals and programmed signals. Any digital output signal the robot has or a robot output signal can be used as the digital output signal.

Each specified digital output signal can be turned on and off under the following conditions:

- Jog signals
- Programmed signals

The same information can be assigned to a programmed signal and a jog signal of the same motion group. In this case, the two signals are logically OR-ed. This means that if either signal is on, the OR-ed signal is also on. It becomes off only when both signals are off.

Jog Signals

All jog signals are turned off when the teach pendant is disabled. If the teach pendant is enabled, only the signals corresponding to the motion group selected by the teach pendant are turned on; the other signals are turned off.

Programmed Signals

Regardless of whether the teach pendant is enabled, programmed signals might be turned on depending on whether the motion group specified by the program that is running or paused; the signals will not be turned on merely by selecting a program. If a program running or paused in multi-tasking mode has a motion group, the programmed signals for that motion group are also turned on.

**10.9.3
Setup**

Use Procedure 10-10 to set up jog and programmed digital output signals.

Procedure 10-10 Setting up the Motion Group DO Output Function

- Step**
- 1 Press MENUS.
 - 2 Select SETUP
 - 3 Press F1, [TYPE].
 - 4 Select Motion group DO. You will see a screen similar to the following.

Motion group DO			JOINT 10%
GROUP	No.	PROGRAM	JOG
1		RO[1]	RO[2]
2		DO[3]	DO[3]
3		RO[0]	RO[0]
[TYPE]		RO	DO

- 5 To change between RO and DO, move the cursor to the signal you want to change and press either F4, RO, or F5, DO.
- 6 Set the signals as desired.

NOTE If 0 is specified for a signal number, the corresponding signal is unspecified, and no output is produced.

10.9.4 Subprogram and Multi-tasking Execution

This section explains how the motion group DO output function works when a subprogram is called or when multi-tasking mode is used.

The programmed signal is created by logically OR-ing signals from all motion groups specified in the program that is running or paused.

If a program with no motion group calls a subprogram that has a motion group, the signals for the motion group are kept on only when the subprogram is being executed. They will not be turned on when the main program (with no motion group) is selected or executed.

If the main program does not have a motion group, but contains a RUN instruction to activate another program to operate a robot in multi-task mode, merely selecting or running the main program does not turn on the signals for the motion group specified in the program activated using the RUN instruction. The programmed signals are turned on only when the program to operate the robot is activated.

Refer to Figure 10-49 for three example programs.

Figure 10-49. Motion Group DO Output Function Program Examples

```

Program "mainpro":motion mask [*,*,*,*]
1: RUN progA
2: RUN progB
:
Program "progA":motion mask [1,*,*,*]
1: J P[1] 100% FINE
:
Program "progB":motion mask [*,1,*,*]
1: L P[1] 500mm/sec Cnt 100
:

```

Program "mainpro" does not have a motion group. It issues a RUN instruction to activate "progA" and "progB," which both have motion groups. "progA" operates motion group 1, and "progB," motion group 2.

Merely selecting program "mainpro" does not turn on programmed signals of any group.

When line 1 of "mainpro" is executed, "progA" is activated, resulting in the signals of motion group 1 being turned on.

When line 2 of "mainpro" is executed, "progB" is activated, resulting in the signals of motion group 2 being turned on.

When "progA" ends, the signals of motion group 1 are turned off. When "progB" ends, the signals of motion group 2 are turned off.

10.10 MOTION START DELAY DETECTION FUNCTION

When a robot must repeat motions within a short time (for example, because of short travel distance or high motion speed), the motions might be decelerated or oscillate even if the termination type of the corresponding motion instruction is “CNT 100.” This is because the robot might not be ready for the next motion when one short motion ends, and cannot start immediately.

The motion start delay detection function detects and displays this kind of a delay in motion start. If deceleration or oscillation occurs, it is possible to determine the program line to be corrected according to the information detected by this function.

The motion start delay detection function is always enabled. The detected information is available at any time.

10.10.1 Restrictions

Keep in mind the following restrictions when using the motion start delay detection function.

- If there is a non-motion instruction (such as a register, I/O, or wait instruction) between motion instructions, it might prevent an immediate start of motion even if the motion instructions use a termination type of CNT100, possibly causing a motion start delay to be detected.
- A CNT termination type setting is ignored when the tool shifts from joint motion to linear or circular motion, or vice versa. The robot always decelerates at such points. Therefore, a motion start delay is detected if the CNT termination type has been specified.
- No motion start delay is detected during single-step execution.

10.10.2 Detection Information and System Variables

The motion start delay detection function detects a motion start delay if the following condition is satisfied:

Condition: When a motion instruction that uses CNT 100 ends, the next motion cannot be started immediately.

If a delay is detected, the related information is recorded in a system variable.

All information related to a motion start delay is recorded in system variable **\$PODATA_GRP[group]**. This system variable is provided for individual motion groups. The contents of the system variable are all cleared at cold start. The fields of this system variable are described in the following sections.

\$PODATA_GRP[.]\$OVERRUN_CNT

This system variable indicates the number of times motion start delays are detected.

\$PODATA_GRP[.]\$CUR_INDEX

This system variable indicates the index of the latest motion start delay data. The information about motion start delays are recorded in 50 memory locations (called buffers) in the order of indexes (sequence number).

Each buffer has a ring structure. When it becomes full of 50 (fixed) pieces of data, the latest data is overwritten to the oldest data. See Figure 10-50.

This system variable holds the index of the information about the motion start delay that occurred most recently. First check this index, then the contents (program ID, program line number, and amount of delay, described later) of the buffer that corresponds to this index.

\$PODATA_GRP[.]\$PROGRAM_ID[50]

This is a program ID memory buffer for a detected motion start delay. It indicates the internal ID of a program encountered with the motion start delay.

\$PODATA_GRP[.]\$LINE_NO[50]

This is a program line number memory buffer for a detected motion start delay. It indicates the program line where the motion start delay was detected.

Keep in mind that the line number indicated here only correspond to the line where a motion start delay was detected. The line that caused the delay has a number lower than this line.

\$PODATA_GRP[.].\$OVERRUN_ITP[50]

This is a motion start delay memory buffer. It indicates the amount of a detected delay. The larger the value, the larger the delay.

If you reference these buffers with the same index (in \$PODATA_GRP[.].\$CUR_INDEX), you can identify the following information:

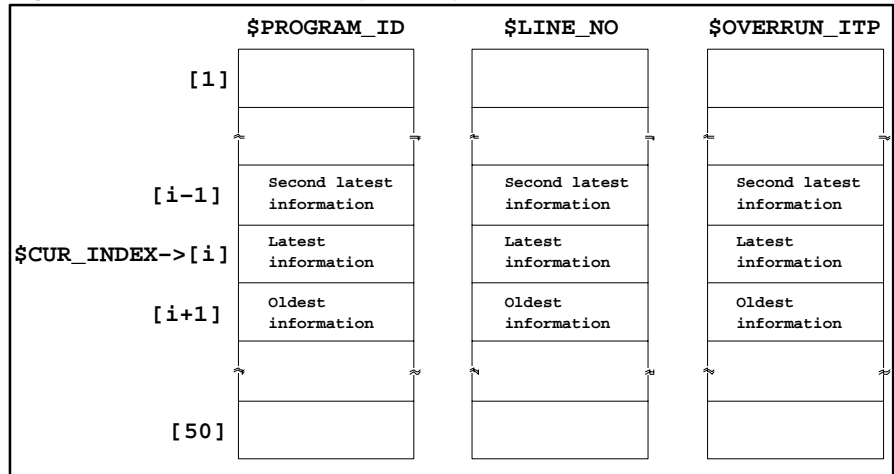
- Which program (\$PROGRAM_ID[i])
- Which motion, or program line (\$LINE_NO[i])
- How much delay (\$OVERRUN_ITP[i])

where “i” is an index.

Each memory area consists of 50 buffer locations. It is always possible to check up to 50 pieces of the latest information about motion start delays.

The memory buffers are shown in Figure 10-50.

Figure 10-50. Motion Start Delay Memory Buffers



10.10.3 Motion Start Delay Detection Function Example

The following sequence describes an example of how to use the motion start delay detection function.

- 1 Test run a program given to the robot to assure safety.
- 2 Reset the following system variables. The function can be used without resetting. However, resetting makes the data more readable.

```
$PODATA_GRP[ ].$OVERRUN_CNT = 0
$PODATA_GRP[ ].$CUR_INDEX = 0
```

- 3 Run the program with the override for actual operation applied. Consider the following example program.

Example: Program (internal ID: 1290)

```
1: J P[1] 100% Cnt 100
2: L P[2] 1000mm/sec Cnt 100
3: L P[3] 1000mm/sec Cnt 100
4: L P[4] 1000mm/sec Cnt 100
5: L P[5] 1000mm/sec Cnt 100
[END]
```

- 4 Check the results of detection. Display the SYSTEM Variables screen, move the cursor to \$PODATA_GRP[], and press the ENTER to view the contents of the system variables.

Assume the following conditions:

Example: Conditions immediately after the above program is executed

```
$OVERRUN_CNT          1
$CUR_INDEX            1
$PROGRAM_ID[1]       1290
$LINE_NO[1]          4
$OVERRUN_ITP[1]      1
```

This example means that a motion start delay occurred when line 4 in a program with internal ID 1290 was executed.

- 5 Correct the program. According to the detected result, it is likely that motions before line 4 (P[4]) might be too close to each other. Check the positional data and correct the position or speed as required.
- 6 Repeat Steps 1 through 5 until deceleration or oscillation does not occur any more.

10.11 SOFT FLOAT FUNCTION (OPTION)

The *soft float function* is a feature that is used to compensate for variances in workpiece precision in applications where the robot is used to mount workpieces on a machine tool.

NOTE You must have the DSP V module to use the soft float function.

In most applications, the robot moves accurately to the taught points. When the robot is used to mount workpieces on a machine tool, variances in workpiece precision can cause the workpiece position to shift relative to the tool. This shifting could result in interference between the workpiece and the tool. The soft float function can compensate for these variances.

The soft float function can be programmed in Joint or Cartesian modes:

- In **Joint mode**, flexibility is specified for individual axes or a combination of axes.
- In **Cartesian mode**, the flexibility, or *softness*, is specified for Cartesian directions. In this mode, the robot will act like a spring in the specified direction, which can be specified in the WORLD, TOOL, or USER coordinate frames.

The soft float function is enabled and disabled using an instruction in a teach pendant program. Its conditions are also specified using the instruction.

To use the soft float function, you must do the following:

- Set **payload information**. Accurate payload information is crucial in getting the desired result with the soft float function. Refer to Section 3.20.
- Set up **soft float schedules**. Setting up schedules involves choosing between Cartesian and Joint mode for each schedule, as well as specifying the flexibility. Refer to Section 10.11.2 for details on setting up soft float schedules.
- Add **soft float instructions** to your teach pendant program. For a description of soft float teach pendant instructions, refer to Section 10.11.3.

Servo Flexibility

When you use the soft float function, you can specify the *servo flexibility* of each axis. Servo flexibility indicates how strongly the axis resists external forces. It is specified between 0% and 100%. A servo flexibility of 0% means the axis will resist external forces very stiffly, but the axis will still move. A servo flexibility of 100% corresponds to being the most flexible. The servo flexibility is specified using a condition 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 can 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 a static load is applied to the robot, the robot controls the force to maintain its attitude even if the soft float function is enabled.

10.11.1 Soft Float Function Restrictions

When you use the soft float function, keep in mind the following restrictions:



WARNING

When the soft float function is enabled,

- **The robot might not always follow the taught path**
- **The taught path changes according to the override**
- **Execution time might be increased**

Be aware of these effects before you enable the soft float function. Otherwise, you could injure personnel or damage equipment.



WARNING

When the soft float function is enabled, robot brakes are affected as follows:

- **If an attempt is made to enable soft float with a brake enabled, the brake is released automatically before the function is enabled.**
- **When the soft float function is enabled, brake control is ineffective.**

Be aware of these effects before you use the soft float function. Otherwise, you could injure personnel or damage equipment.

- The soft float function is **disabled automatically** when
 - Program execution starts.
 - Program execution ends.
 - The program stops due to an alarm that turns off the servomotors.
 - Jogging is performed with the program paused.
 - The program is restarted after the cursor is moved manually with the program paused.
 - Backward execution is performed.
 - Power is turned on.
- If the **program is paused and then restarted**, the states of the soft float function (such as enabled/disabled and the exec start ratio) are set to the conditions that existed before the program was paused, except for the cases listed in the previous item that cause the soft float function to be disabled automatically.

- The soft float function **cannot be enabled** by any method other than the SOFTFLOAT instruction.
- When the **soft float function is enabled**, the robot moves using the termination type CNT0 (no position check is made), even if FINE has been specified as motion instruction termination type.
- When the **soft float function is enabled**, if an external force causes the robot to move beyond a certain distance, the following servo error messages are displayed:
 - If the robot is at rest: [SRVO-023 Stop error excess(G:i A:j)]
 - If the robot is operating: [SRVO-024 Move error excess(G:i A:j)]
- If an attempt is made to **enable the soft float function with a brake applied**, the brake is released automatically before the function is enabled.
- When the **soft float function is enabled**, brake control is ineffective.
- If the **motion group mask in a program is [*,*,*,*]** (there is no motion group), when the program issues instructions with the soft float function, the following error message is displayed:
[INTP-216 (program name, line number) Invalid value for group number]
- Minimize the **range of motion** used with the soft float function enabled.

Load distribution on the robot's axes varies with the position and speed of the robot. When the robot is moved away from the nominal position (at which the soft float function was enabled), these variations can cause the robot to move in a direction other than the direction of the applied external force. Since gravity is the most dominant load, this motion is often along the z direction. As a result of this behavior, you should try to keep the range and speed of robot motion to a minimum when the soft float function is enabled.

- When the soft float function is enabled, if follow-up processing requires more time than specified in system variable \$SFLT_FUPTIM, the servo alarm or program pause alarm occurs. The system variable \$SFLT_ERRTYP specifies which alarm will occur.

\$SFLT_FUPTIM Default value: 1000 (ms)

This value varies from one system to another. The large value that does not cause an alarm during normal operation should be used.

\$SFLT_ERRTYP Default value: 0

–If 0, servo alarm "SRVO-111 Softfloat time out" occurs.

–If 1, Program pause alarm "SRVO-112 Softfloat time out" occurs.

(The alarm number is different between the alarms.)

The default value should be used unless turning the servo off invites any inconvenience in the system.

Refer to the FOLLOW UP teach pendant instruction in Section 10.11.3 for more information on follow-up processing.

- When the soft float function is enabled, follow-up processing is normally performed for individual motion instructions. This processing is enabled or disabled according to system variable \$SFLT_DISFUP.

\$SFLT_DISFUP Default value: FALSE

- If FALSE, follow-up is performed at the start of each motion instruction in the program.
- If TRUE, follow-up is not performed for individual motion instructions in the program.

Refer to the FOLLOW UP teach pendant instruction in Section 10.11.3 for more information on follow-up processing.

10.11.2 Soft Float Schedules

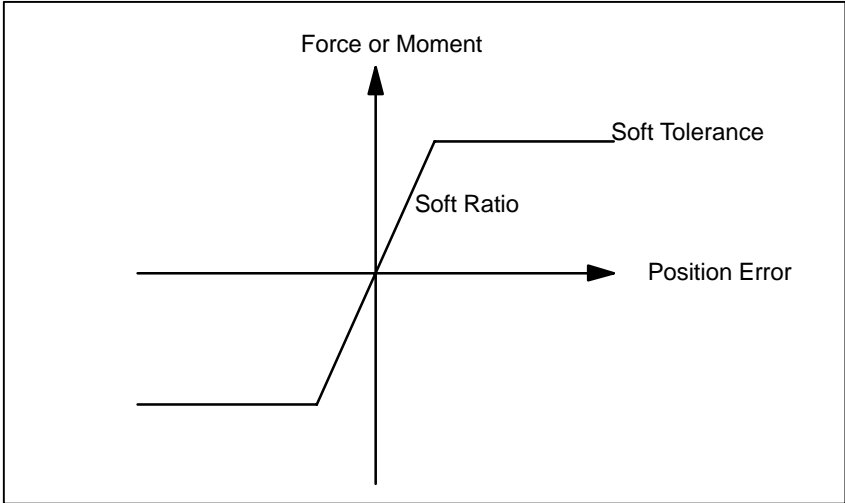
You specify soft float schedules on the SETUP Softfloat screen. You can specify up to ten schedules for the soft float function.

Table 10-13 lists and describes each item soft float schedule item on the SETUP Softfloat screen.

Table 10-13. Soft Float Schedule Setup Items

ITEM	DESCRIPTION
DETAIL Software (Joint) Screen Items	
Comment	Pressing the ENTER key with the cursor on line 1 enables entering a comment. Press the appropriate function keys to enter the comment.
Exec Start Ratio	Exec start ratio specifies the point where the soft float function is enabled if the SOFTFLOAT[n] instruction is used as a motion option. Refer to the motion option section of Section 10.11.3 for more information on the exec start ratio.
Axisn Soft Ratio (Servo Flexibility)	Servo flexibility for each axis (n) can be specified on the third line of the DETAIL screen and on subsequent lines. The servo flexibility (softness ratio) indicates how strongly the axis resists external forces. It is specified between 0% and 100%. A flexibility of 100% corresponds to being the most flexible.
Enabled/disabled	Specifies whether the soft float function is enabled or disabled for each axis. Specify this on line 3 of the DETAIL screen and on subsequent lines. Setting the cursor at the rightmost end (enabled/disabled setting position) of each line causes the F4 (ENABLE) and F5 (DISABLE) keys to appear. Use these keys to specify whether to enable/disable the soft float function.
DETAIL Softfloat (Cartesian) Screen Items	
Schedule No.	Specifies the schedule number. By default, you can set up ten schedules.
Enable/disable	Specifies whether to enable or disable the Cartesian soft float function. When set to DISABLE, soft float will not be executed.

Table 10-13. (Cont'd) Soft Float Schedule Setup Items

ITEM	DESCRIPTION
Coordinate	Specifies the frame used: World, User, or Tool
X direction Y direction Z direction X rotation Y rotation Z rotation	<p>Specifies the softness (flexibility) in the X, Y and Z axes directions and rotations. When Soft Ratio is enlarged, the spring constant in the direction will become smaller. When Soft Tolerance is enlarged, maximum of the force and the moment put out in the direction will be reduced.</p> 

Use Procedure 10-11 to set up Joint and Cartesian soft float schedules.

Procedure 10-11 Setting Up Soft Float Schedules

Condition ■ You have set up payload information properly. Refer to Section 3.20.

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select Softfloat. See the following screen for an example of the listing screen.

SETUP SOFTFLOAT					JOINT 30%	
Group 1						1/10
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 To change the display from detail to listing, press F3, LIST.

To change the display from listing to detail, press F3, DETAIL.

Cartesian Soft Float Schedules

6 To set up schedules for Cartesian soft float, press F3, DETAIL, and then press F5, CART. See the following screen for an example.

SOFTFLOAT (CARTESIAN)				
Group 1				
1	Schedule No[1]:[]	
2	Enable Disable	:	[DISABLE]	
3	Coordinate	:	[WORLD]	
		Soft Rat	Soft Tol	
4	X direction:	[0]%	[0]%	
5	Y direction:	[0]%	[0]%	
6	Z direction:	[0]%	[0]%	
7	X rotation:	[0]%	[0]%	
8	Y rotation:	[0]%	[0]%	
9	Z rotation:	[0]%	[0]%	
[TYPE] NUMBER LIST JOINT CART >				
GROUP LIST >				

- 7 To enable or disable the soft float function for a direction, move the cursor to the word “ENABLE” or “DISABLE” and press F4, ENABLE, or F5, DISABLE.
- 8 Set the coordinate system to the desired value.
- 9 Set the softness ratio and softness tolerance to the desired values.
- 10 To select another page of the detail screen for other schedules, press F2, NUMBER.

Joint Soft Float Schedules

- 11 To set up Joint Soft Float schedules, move the cursor to the schedule you want to set up and press F3, DETAIL, and then press F4, JOINT. See the following screen for an example.

SETUP / SOFTFLOAT		JOINT 30%	
Group 1			
1	Schedule No[1]:	*****	
2	Exec Start Ratio :	0 %	
3	Axis1 Soft Ratio :	0 %	DISABLE
4	Axis2 Soft Ratio :	0 %	DISABLE
5	Axis3 Soft Ratio :	0 %	DISABLE
6	Axis4 Soft Ratio :	0 %	DISABLE
7	Axis5 Soft Ratio :	0 %	DISABLE
8	Axis6 Soft Ratio :	0 %	DISABLE
9	Axis7 Soft Ratio :	0 %	DISABLE
10	Axis8 Soft Ratio :	0 %	DISABLE
11	Axis9 Soft Ratio :	0 %	DISABLE
[TYPE]	NUMBER	LIST	JOINT CART >
GROUP		LIST	>

- 12 To enable or disable the soft float function for an axis, move the cursor to the word “ENABLE” or “DISABLE” on the line that corresponds to the axis you want and press F4, ENABLE, or F5, DISABLE.
- 13 Set the exec start ratio and the appropriate softness ratios to the desired values.
- 14 To select another page of the detail screen for other schedules, press F2, NUMBER.

10.11.3 Soft Float Program Instructions

The following soft float instructions can be used:

- SOFTFLOAT[n]
- SOFTFLOAT END
- FOLLOW UP

SOFTFLOAT[n]

SOFTFLOAT[n] enables the soft float function from the next instruction in your teach pendant program until a SOFTFLOAT END instruction is executed. The soft float function enabled will use soft float schedule “n.”

Refer to Section 10.11.2 to define soft float schedules.

NOTE You must set up payload information properly and specify the correct payload schedule in the program to use the soft float function. Accurate payload information is crucial in getting the desired result with the soft float function. Refer to Section 3.20.

You can use SOFTFLOAT[n] in two forms within your teach pendant program. The form you use determines when during program execution the soft float function will be active. The SOFTFLOAT[n] instruction can be specified:

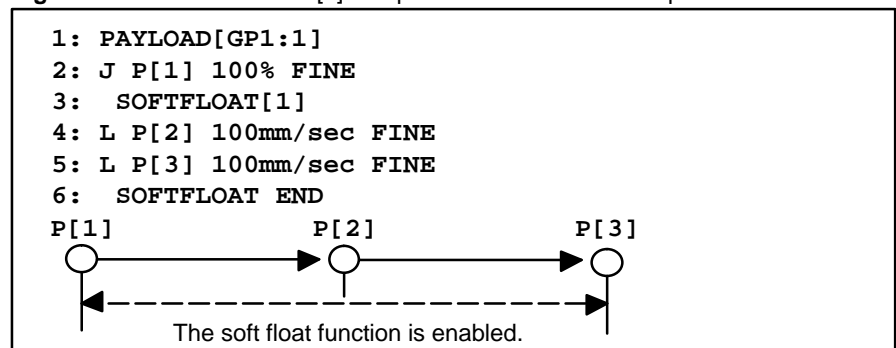
- As an **independent instruction**, on its own line within the program.
- As a **motion option** within a motion instruction.

Independent Instruction SOFTFLOAT[n]

When specified as an **independent instruction** on its own line within the program, the soft float function is enabled after the end of the motion specified on the line preceding the independent SOFTFLOAT[n] instruction.

In Figure 10-51, the soft float function is enabled after the motion specified on line 1 completes, and disabled using SOFTFLOAT END on line 5. In this example, the FINE termination type in lines 4 and 5 has the same effect as CNT0.

Figure 10-51. SOFTFLOAT[n] Independent Instruction Example



Motion Option L P[i] 100mm/sec FINE SOFTFLOAT[n]

When specified as a **motion option** within a motion instruction, the soft float function becomes enabled during the execution of the motion statement that contains the SOFTFLOAT[n] motion option.

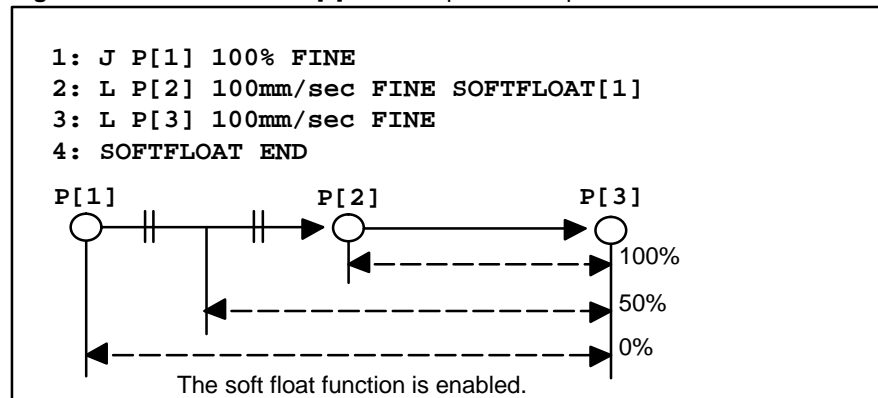
The point at which the soft float function becomes enabled is determined by a soft float schedule item, Exec Start Ratio. This is available only for JOINT soft float schedules.

The exec start ratio is specified as the percentage (from 0% to 100% in 1% steps) of a distance to be traveled before the robot reaches the taught point that corresponds to the motion instruction that contains the SOFTFLOAT[n] motion option.

In Figure 10-52, the soft float function is effective between P[1] taught using the motion instruction on line 1, and P[2] taught using the motion instruction on line 2 that contains the SOFTFLOAT[n] motion option. In this example, the FINE termination type in line 3 has the same effect as CNT0.

- **Exec Start Ratio 100%** – The robot has completed 100% of the move to P[2] when soft float is enabled. This means that the robot is at P[2].
- **Exec Start Ratio 50%** – The robot has completed 50% of the move to P[2] when soft float is enabled. This means that the robot is halfway between P[1] and P[2].

Figure 10-52. SOFTFLOAT[n] Motion Option Example



SOFTFLOAT END

SOFTFLOAT END disables the soft float function. Add this instruction within your teach pendant program at the location where you want to stop using the soft float function.

FOLLOW UP

When an external force is removed from a robot, the robot usually tries to go back to the taught point. However, the FOLLOW UP instruction causes the robot to assume that the current position is the taught point, and prevents it from going back to the taught point. Use this instruction to override the robot's usual behavior when an external force is removed.

10.12 CONTINUOUS TURN FUNCTION

Previously, one of the axes of a robot could not be rotated in a given direction continuously and indefinitely. The continuous turn function allows the last axis and extended rotation axis of the robot to turn in a given direction continuously and indefinitely. For example, the last axis for a robot with six axes is J6.

This function can be used when a part that requires continuous rotation, such as a conveyer, pump, or grinder, is to be operated by a robot or its rotation axis.

This function is enabled or disabled by using the newly provided [SETUP Cont turn] screen. Continuous rotation is started and stopped by the program.

Before this function can be used, the settings necessary for continuous rotation must be made.

The continuous turn axis can be assigned to an axis that satisfies any one of the following conditions, and only one axis can be assigned for each motion group:

- Last axis of the robot
- Last extended axis of the integrated rotation axes
- Any extended axis of the auxiliary rotation axes
- Last axis of the Nobot

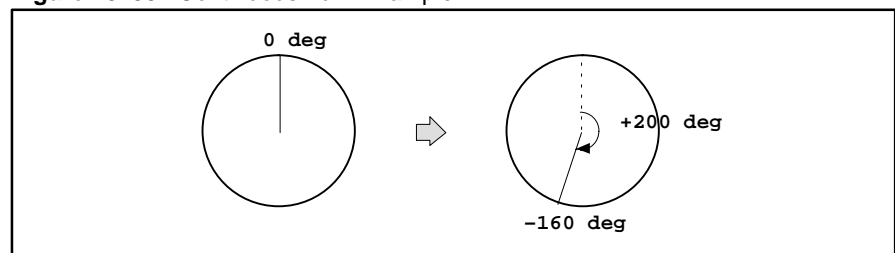
The continuous turn axis must satisfy the following mechanical conditions:

- The mechanism must allow continuous rotation. (There must be no obstacles such as stoppers.)
- The gear reduction ratio (the value of [Numerator of Gear Ratio] [Denominator of Gear Ratio] on the setup screen) must be no more than 4000.

10.12.1 Function

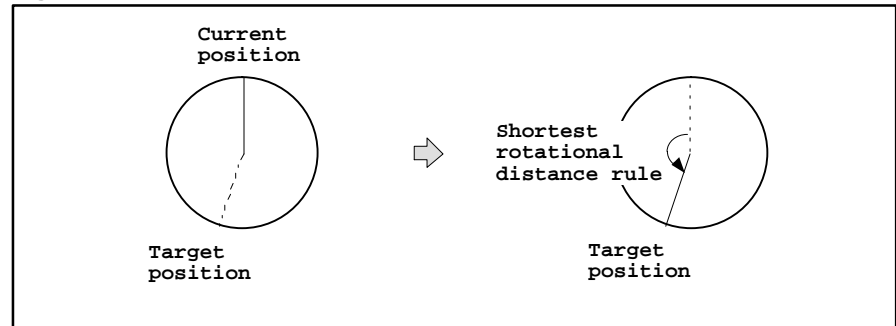
When this function is enabled, the axis assigned as the continuous turn axis turns indefinitely; so, the angle of that axis is indicated as a relative position within $+180\text{deg}$, instead of as an absolute position. The following example shows that the axis turns from the 0deg position to the 200deg position in the positive direction, and that the position after the rotation is indicated as -160deg , not 200deg . See Figure 10-53.

Figure 10-53. Continuous Turn Example



When continuous rotation is not performed with this function enabled (the usage will be explained later), the continuous turn axis turns in whichever rotation direction minimizes the angular displacement from the current position to a target position. Normally, the rotation direction is determined uniquely from the relationship between the target position and current position. This “the Shortest rotational distance rule” is effective in reducing the cycle time. See Figure 10-54.

Figure 10-54. Shortest Rotational Distance Rule



10.12.2 Setting

To use this function:

- Make necessary settings on the [SETUP Continuous turn rotation velocity (CTV)] screen.
- Specify the start and end of continuous rotation by using the continuous turn rotation velocity (CTV) instruction, motion option instruction.

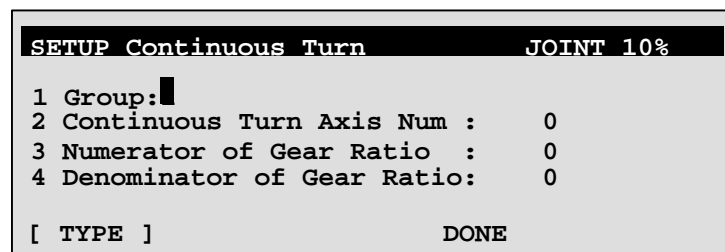
Table 10-14 lists and describes the continuous turn items you must set. Use Procedure 10-12 to set up continuous turn.

Table 10-14. Continuous Turn Setup Items

ITEM	DESCRIPTION
Group	Set a motion group number.
Continuous Turn Axis Number	Set the axis number of the continuous turn axis. When 0 is set, this function is disabled for the indicated motion group.
Numerator of Gear Ratio	Set the gear reduction ratio of the continuous turn axis specified for the above item. Values from 0 to 32766 can be specified. The specified values must satisfy the following: [Numerator of Gear Ratio] / [Denominator of Gear Ratio] ≤ 4000
Denominator of Gear Ratio	

Procedure 10-12 Settings for the Continuous Turn Function

- Step**
- 1 Press the MENU key to display the screen menu.
 - 2 Select SETUP.
 - 3 Press F1 [TYPE] to display the screen switch menu.
 - 4 Select Cont Turn. Then, the continuous turn rotation velocity screen appears.



- 5 Make the necessary settings by using the numeric keys and other keys.
 - To disable the continuous turn function, set 0 for “Continuous Turn Axis Num.”
 - For both “Numerator of Gear Ratio” and “Denominator of Gear Ratio,” enter values of up to 32766.
 - Set a motion group number for “Group.” When the value in this field is changed (the number of a motion group the user wants to display), the settings in the other fields are also changed accordingly to display the values associated with the new motion group.
- 6 After all settings have been completed, press F4, DONE. Then, the following message appears:
[Must COLD start to take effect]
- 7 Turn off the power then turn it on again to cold-start the system. Note that cold start is not selected automatically.

10.12.3 Motion Instruction

After the settings for the continuous turn axis are complete, specify the start point for continuous rotation by using the continuous turn rotation velocity (CTV) instruction, an additional motion instruction.

The following continuous turn rotation velocity instruction is provided. Teach this instruction as an additional motion instruction.

NOTE Teach this instruction the same way you teach other motion instructions. Refer to Chapter 5.

- Continuous turn rotation velocity instruction CTV i
* i = -100 to 100 Turn axis speed, as a percentage of the maximum axis speed (%)

Start of Continuous Rotation

Continuous rotation starts as soon as a motion instruction with a continuous turn rotation velocity instruction is executed.

End of Continuous Rotation

Continuous rotation terminates when the first motion statement that has no continuous turn rotation velocity instruction is encountered after the continuous turn rotation velocity instruction is started.

When continuous rotation is terminated, the other axes belonging to the same motion group are stopped temporarily. Therefore, even when the termination type for the previous motion is Cnt, the robot decelerates. The continuous turn axis starts decelerating after the other axes have entered the stopped state, then stopping at the taught position. At this time, the continuous turn axis is not synchronized with the other axes (including axes belonging to other motion groups).

10.12.4 Operation

The following items describe the operation of continuous turn:

- Continuous rotation is maintained during the execution of logic instructions (other than motion statements).
- When a program is played back, the turn number of the continuous turn axis is ignored, and is always assumed to be 0.
- The turn number of the continuous turn axis at the point touched up when this function is enabled is always recorded as 0.
- When 0 is specified for the turn axis speed in the continuous turn rotation velocity instruction, continuous rotation is not performed, and the continuous turn axis takes the shortest rotational distance rule. Specifying 0 as the turn axis speed in this instruction is a useful means of temporarily stopping the continuous turn axis without stopping the robot (normally, the end of continuous rotation stops the robot temporarily). (See "Example" below.)

- In single-step execution (forward program execution and reverse program execution), continuous rotation is not performed even when a continuous turn rotation velocity instruction is specified, and the axis takes the shortest rotational distance rule.
- A hold stops continuous rotation. When program execution is restarted after the hold, continuous rotation is not performed if axes other than the continuous turn axis are already at the target positions. If they are not at the target positions, continuous rotation is also restarted.
- Continuous rotation can be performed by jog feed.

10.12.5 Example

The following shows an example of using a continuous turn rotation velocity instruction:

```

1: J P[1] 100% FINE
2: J P[2] 100% Cnt100 CTV 100
3: J P[3] 100% FINE
4: J P[4] 100% Cnt100 CTV 100
5: J P[5] 100% FINE CTV 0
6: J P[6] 100% FINE
7: J P[7] 100% FINE CTV 100
8: WAIT 100.0sec
9: J P[8] 100% FINE

```

- Explanation of line 1 to line 3
During a motion from POS[1] to POS[2], continuous rotation is performed. Although termination type Cnt is specified in line 2, the robot decelerates because a continuous turn rotation velocity instruction is not specified in line 3. (All axes are stopped temporarily at the start of motion in line 3.)
- Explanation of line 4 to line 6
As soon as line 4 is executed, continuous rotation starts. Since 0 is specified as the turn axis speed in the continuous turn rotation velocity instruction of line 5, continuous rotation is interrupted upon the start of the execution of line 5. In this case, the continuous rotation state is maintained. So, the termination type Cnt100 in line 4 is effective, and the robot does not decelerate.
When line 5 is executed, the continuous turn axis turns in whichever direction minimizes the angular displacement to a target position.
- Explanation of line 7 to line 9
As soon as the motion in line 7 is started, continuous rotation starts. Continuous rotation is continued when the WAIT instruction (logic instruction) in line 8 is executed.
Upon the start of the motion in line 9, all axes of the robot are stopped temporarily, and continuous rotation is terminated.

10.12.6 Notes and Restrictions

Before using this function, note the following:

- When continuous rotation is performed with a robot axis or integrated extended axis, both the X and Y components in the tool coordinates must be 0. (Only the Z-axis component can hold a non-0 value.) If this requirement is not satisfied, the path of linear or circular motion will be unpredictable in normal operation other than continuous rotation.
- This function cannot be used together with the following functions:
 - Independent extended axis velocity instruction (The dependent extended axis velocity instruction can be used with the continuous turn function.)
- This function automatically updates the mastering data (only for the continuous turn axis) according to the number of turns the continuous turn axis rotates. So, the recorded mastering data may not match the current mastering data. When this function is disabled, mastering need not be performed again.
- When this function is disabled, the current position of the continuous turn axis may exceed the stroke limit. If this occurs, move the continuous turn axis to within its stroke limits by using jog feed or the program.
- In a multi-group system, when the settings on the [SETUP Cont Turn] screen are modified, then the F4 [DONE] key is pressed, system variable \$PARAM_GROUP[group].\$SV_OFF_ENBL[i] (i: axis number) must be set to FALSE for all axes of all motion groups to disable brake control, before the power is turned off then on again to cold-start the system.
- When more than one continuous turn axis is specified in a multi-group system, different continuous rotation speeds cannot be specified for these axes.
- When continuous rotation is terminated, the continuous turn axis gradually decelerates to stop, such that the axis may rotate one or more turns (the number of turns the axis rotates varies according to the acceleration/deceleration time constant.)
- Also, in backward program execution (performed in single step mode), the continuous turn axis takes the shortest rotational distance. If forward step execution and backward program execution are executed successively for a motion statement with an angular displacement very close to 180deg, the continuous turn axis may turn in the same direction throughout both forward execution and reverse execution.

10.12.7 Alarm Codes

The alarm codes related to this function are as follows:

CNTR-001 WARN No global variables

Cause: Continuous Turn global variables are not loaded.

Remedy: Perform a controlled start and initialize motion softparts.

CNTR-002 WARN No MIR pointer

Cause: This is an internal system error.

Remedy: Perform a cold start on the controller.

CNTR-003 WARN WARN No sysvar pointer

Cause: This is an internal system error.

Remedy: Perform a cold start on the controller.

CNTR-004 WARN No cnir pointer

Cause: This is an internal system error.

Remedy: Perform a controlled start and initialize the motion softparts.

CNTR-005 WARN WARN Illegal Cont. Turn Axis

Cause: The continuous turn axis that was selected is not a valid cont. turn axis, or cn_gear_n1 or cn_gear_n2 have a zero value

Remedy: Check Continuous turn axis, cn_gear_n1, and cn_gear_n2 Select different continuous turn axis and/or set correct gear ratio for continuous turn axis.

CNTR-006 WARN WARN Unable to Allocate Memory

Cause: A failure occurred while allocating memory.

Remedy: Check amount of memory being used by the system.

CNTR-007 STOP.G STOP.G Serious Internal error (G:%d^2)

Cause: Internal Continuous Turn error

Remedy: Record error and report to hotline

CNTR-008 STOP.G Invalid dest. angle, (G:%d^2)

Cause: Invalid destination angle during linear motion. Incompatibility with Continuous Turn and other options.

Remedy: Check compatibility of motion options. Remove other options.

CNTR-009 WARN WARN Warn-Cont Vel too high(G:%d^2)

Cause: Continuous turn axis velocity is too high. cn_turn_no will not be valid because of high rotational speed.

Remedy: Lower contaxisvel. This warning may be ignored if cn_turn_no is not used.

CNTR-010 STOP.G STOP.G Ind.EV option not allowed.

Cause: Continuous turn is not compatible with independent extended axes The Ind.EV motion option is not allowed.

Remedy: Remove Ind.EV option or disable continuous turn on the group

CNTR-011 STOP.G STOP.G Axis speed exceeds lim(G:%d^2)

Cause: Programmed motion exceeds the speed limits on the continuous turn axis. Speed limit is 180 degrees per ITP time

Remedy: Lower the speed either through KAREL or Teach Pendant

CNTR-012 STOP.G STOP.G Ending Cont Rot on Rel Motion

Cause: Attempted to end Continuous Rotation with a Relative Motion

Remedy: Continuous Rotation must be ended with an absolute motion Use an absolute motion to end continuous rotation

10.13 CRT FUNCTION

The CRT function enables a factory terminal to be connected to the R-J2 controller. The factory terminal can provide almost the same display as a teach pendant and enables using a full keyboard.

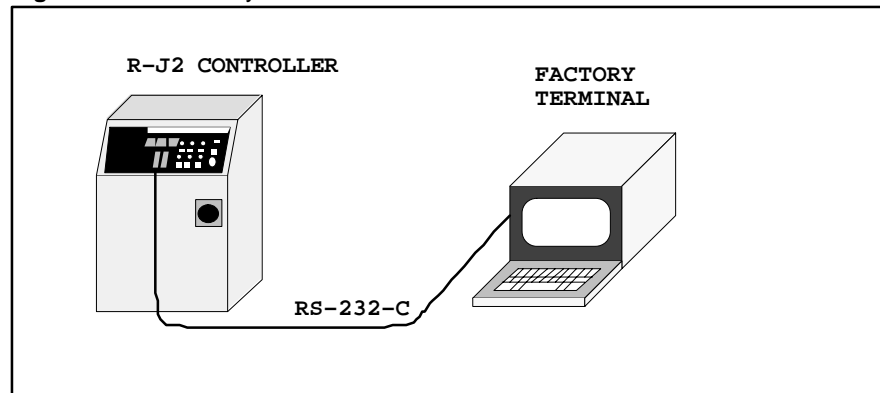
Connecting the R-J2 controller to a factory terminal requires a CRT connection option (A05B-2350-J535) as well as the following:

- Factory terminal A13B-0144-B002 * B001 cannot be used.
- Interconnection cable (3 or 7 m) A13B-0144-K001/002
- RAM module with a memory capacity of at least 2M bytes

An RS-232-C cable is used to connect the R-J2 control unit to the factory terminal. Power to the terminal is supplied from the R-J2 control unit through the RS-232-C cable. See Figure 10-55.

NOTE In North American HandlingTool, the CRT function is available only on a VT-220 compatible device.

Figure 10-55. Factory Terminal



The CRT function enables much the same operations as with the teach pendant. However, some operations (such as operating the robot directly) are prohibited in order to assure safety.

For operations possible with this function, refer to Section 10.13.1.

To connect a factory terminal to the R-J2 controller, use Procedure 10-13 .

Procedure 10-13 Connecting a Factory Terminal

- Step**
- 1** Turn off the controller.
 - 2** Connect the RS-232-C cable from the factory terminal to the port on the R-J2 controller.
 - 3** Cold start the controller.
 - 4** To set up the port to use the factory terminal, perform the following steps.
 - a** Press **MENU**.
 - b** Select **[SETUP]**.
 - c** Press **F1**, **[TYPE]**.
 - d** Select **Port Init**. A list of ports appears on the screen.
 - e** Move the cursor to the desired port number and press **F3**, **[DETAIL]**. The port setup screen appears.
 - f** To set up the communication unit, move the cursor to **Device** and press **F4**, **[CHOICE]**. Select an item corresponding to the terminal to be used.
Factory terminal → “**FACTORY TERMINAL**”
VT220 or compatible terminal → “**KCL/CRT**”

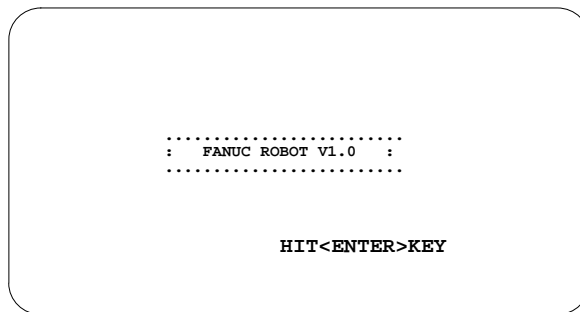
When the communication unit is selected, the other setup fields are set to the default values automatically.
 - g** When you are finished setting up the port, press **F2**, **[LIST]**, to resume the list of ports.
 - h** Turn off the controller.
 - 5** If you change a port setting connected to a **KCL/CRT** or reconnect a factory terminal to another port, you must perform a cold start.

NOTE The factory terminal does not need to be on to perform this procedure.

Use Procedure 10-14 to start the factory terminal.

Procedure 10-14 Starting the Factory Terminal

- Step**
- 1 Make sure the controller and the factory terminal are connected. Turn on the factory terminal. At this point, power is not supplied to the terminal, because the R-J2 controller unit has not been turned on.
 - 2 Cold start the controller. On the factory terminal, you will see a screen similar to the following.



NOTE If this display does not appear, and a light beside the LOCK key on the factory terminal is blinking, it is likely that the connection is incomplete. Turn off the R-J2 controller, check the connection, and return to Step 1.

- 3 After the hints menu appears on the teach pendant, press the ENTER key on the factory terminal.
- 4 After the message “HIT <ENTER> KEY” disappears, press the TAB key on the factory terminal.
- 5 After the factory terminal is activated, either of the following menus appears:
 - If cold-started, the hints menu
 - If hot-started, the screen that was on the teach pendant when the power was switched off

NOTE If the TAB key is pressed too soon in Step 3, all other keys might be disabled. If such a failure occurs, press the TAB key again.

NOTE The TAB key on the factory terminal should be pressed slowly and firmly. If a light beside the LOCK key blinks, or the factory terminal does not respond when the F10 key is pressed, turn off the controller unit and repeat the above procedure from the beginning.

10.13.1 Operation

After the factory terminal has been started, the same display as the teach pendant appears on the CRT screen, and the full keyboard is available. See the following screen for an example.

Example of Display

```

UTILITIES Hints

MENUS - Provides a list of all menus.
FCTN - Provides helpful functions.
SELECT, EDIT, DATA, POSN, I/O, STAT -
  Displays menus with those names.
Function keys with [ ] show more choices.
F1 [ TYPE ] function key lists related
  screens within a menu.

[ TYPE ]
|  F1  ||  F2  ||  F3  ||  F4  ||  F5  |
-----
| CURSOR   SCR UP   SCR DN   FCTN   MENUS
|  F6  ||  F7  ||  F8  ||  F9  ||  F10 |

```

When operating a factory terminal, observe the following:

- To edit programs on the factory terminal, it is necessary to keep the teach pendant enable switch of the teach pendant turned on.
- You cannot execute a program or jog the robot from a factory terminal.
- The following items apply to displaying screens on the teach pendant and factory terminal:
 - You cannot display the same screen on the factory terminal and teach pendant automatically.
 - You can display the same screen on the factory terminal and teach pendant if you display the screen on each device manually.
 - Screen changes are not updated dynamically. If you are displaying the same screen on both devices, you must leave and re-enter the screen to update changes.
 - If a program editing menu is opened on both devices simultaneously, the movement of the cursor on one device might affect that of the other device. In such a case, exit the program editing menu on one of the devices.

Table 10-15 lists the operations on the teach pendant and the corresponding operations on the factory terminal.

Table 10-15. Operations on the Teach Pendant and the Corresponding Operations on the Factory Terminal

Function	Teach pendant	Factory terminal
Selection of F1 to F5	F1 to F5	F1 to F5
Moving back	Moving back (PREV)	Key at the left of F1
Selection of the next operation	NEXT	Key at the right of F10
SHIFT	SHIFT	SHIFT (effective for alphanumeric characters including numerals and symbols)
Menu selection	MENU	F10, MENUS
Program list	SELECT	Press the following in the stated order: F10 MENUS 0 — NEXT — 1 SELECT
Program editing	EDIT	Press the following in the stated order: F10 MENUS 0 — NEXT — 2 EDIT
Data (register) display	DATA	Press the following in the stated order: F10 MENUS 0 — NEXT — 3 DATA
Miscellaneous function selection	FCTN	F9, FCTN
Cursor movement	Cursor keys	Cursor keys
Cancellation	BACK SPACE	BS
Items	ITEM	F6, CURSOR
Data entry	ENTER	ENTER
CAPS LOCK (uppercase letters)	None	LOCK
Half page up	SHIFT + ↑	F7, SCR UP
Half page down	SHIFT + ↓	F8, SCR DN

For safety reasons, the factory terminal does not have keys that correspond to the following operations:

- Forward/backward motion (FWD/BWD)
- Manual feed coordinate system (COORD)
- +%/-% (+%/-%)
- Hold (HOLD)
- Jog (JOG KEYS)
- Step (STEP)
- Reset (RESET)
- Positional correction on the program editing screen using SHIFT + F5
- Simultaneous deletion of all alarm history data on the alarm screen using SHIFT + F4
- Simultaneous deletion of all macros on the macro setup screen using SHIFT + F2
- Braking on the manual brake screen using SHIFT + F2 or SHIFT + F3
- Simultaneous use of the SHIFT key with another key

The following keys on the factory terminal are disabled:

- CTRL
- hold screen
- → →
- ← ←
- CANCEL CMND

The following keys on the factory terminal have special functions:

- ESC
Pressing the ESC key disables the keyboard except for the F1 to F10 function keys. Pressing any key from F1 to F10 enables the keyboard again.
- TAB
Pressing the TAB key causes the R-J2 control unit to redisplay a screen on the factory terminal and teach pendant. Be careful not to press the TAB key more than two times in a row.

10.13.2 Troubleshooting

When you use the CRT, you might encounter the following problems:

- No display appears on the factory terminal.
 - Check the connection between the R-J2 control unit and the factory terminal.
 - Check the power to the controller and the power switch of the factory terminal.
 - Check the setting of the port used.
 - Press the TAB key to redisplay a menu.
 - If a partial menu appears, press the TAB key to redisplay the screen.
- The message “HIT <ENTER> KEY” does not disappear from the factory terminal.
 - When power is supplied to the controller, press the ENTER key on the factory terminal. Press the ENTER key before the TAB key.
 - If it is impossible to operate the factory terminal, check the setting of the port used and the connection between the controller and factory terminal, then perform Procedure 10-14 .
- After the factory terminal has been started, either no display appears on the factory terminal screen at all, or part of it is missing.
 - This failure probably occurs if the TAB key on the factory terminal has been pressed too soon after the controller has been turned on, if the power switch of the factory terminal is set to on after the controller has been turned on, or if there is an intermittent problem with the cable connection. If pressing TAB again does not cause a complete display to appear, perform Procedure 10-14 .
- The factory terminal does not respond when a key is pressed.
 - The ESC key might have been pressed accidentally. Press any key from F1 to F10 to resume normal operation.
- The cable has been detached during use.
 - If the cable has come off during operation, switch off the factory terminal and attach the cable again. After supplying power to the factory terminal, make sure that the light beside the LOCK key does not blink. If the light does not blink, the cable is connected successfully. Press the TAB key to resume operation. If the light blinks, detach the connector and attach it again.
- Programs cannot be edited.
 - When editing programs on the factory terminal, it is necessary to turn on the teach pendant.
- A message is not displayed on the user screen.
 - The factory terminal user screen is designed so that message instructions are not displayed on it. Use the teach pendant user screen to display user messages.

10.14 SINGULARITY CHECK FUNCTION

When position data is used in Cartesian coordinates to perform motion instruction teaching or position modification, the robot position might be close to a singular point. In such a case, when the taught motion instruction is executed, the robot can operate at an attitude other than that with which the instruction was taught.

To prevent this, the singularity check function is provided. When a position is taught, this function checks whether the taught position is at a singular position. The function can record the position in joint coordinate format, if so specified by the user.

Function

To enable the singularity check function, set system variable \$MNSING_CHK to TRUE.

If a motion instruction is taught using the SHIFT and POINT keys, or a position is modified by using the SHIFT and TOUCHUP> keys, while the robot is at a singular point, this function checks whether the taught position is at a singular point.

This check is performed when all of the following conditions are satisfied:

- The position is recorded in Cartesian coordinates.
- An additional instruction with INC or offset is not specified.
- The UF (user coordinate system number) of the position data is 0.

If the check finds that the taught position is at a singular point, the following warning message appears in the upper two lines on the teach pendant:

[Can't record on Cartesian (G:i)]

[In singularity]

where, "i" is the number of the motion group at the singular point.

At the same time, the following prompt appears at the bottom on the teach pendant:

[Record current position on joint?]

The [YES] and [NO] function keys are displayed. Select the appropriate key.

- [YES]: The position data is recorded in joint coordinate format.
- [NO]: The position is not taught or modified.

For position data in a program that has multiple motion groups, the singularity check is performed in ascending order of group number. When more than one group is at a singular point, the warning message and prompt message appear repeatedly for each group.

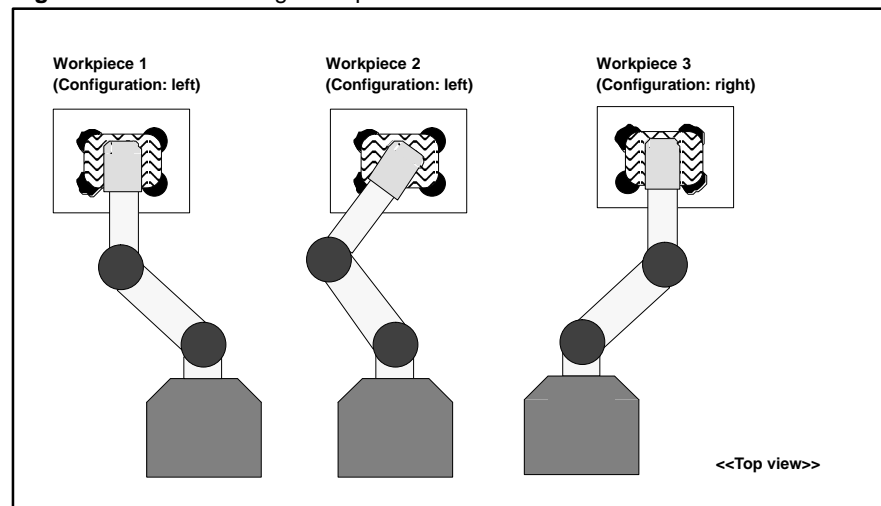
This function is not available for teaching BOTTOM POINTS for the palletizing function or for teaching the ROUTE POINTS.

10.15 ALL-POINT TEACHING FOR PALLETIZING

Conventional robots are capable of palletizing or depalletizing workpieces in a single configuration. Refer to “Positional Information” for information about configuration.

Figure 10-56 shows an example of palletizing with the FANUC Robot M-400 (horizontally articulated robot with two arm configurations, left/right).

Figure 10-56. Palletizing Example



In this example, the palletizing of three workpieces is taught using different configurations—right for workpiece 3 and left for the other two.

Even after teaching as above, in a conventional system the program is executed so that the robot palletizes or depalletizes workpieces such that the configuration is the same at each bottom point.

This function enables a robot to palletize or depalletize workpieces according to the configurations it has been taught, simply by setting a system variable.

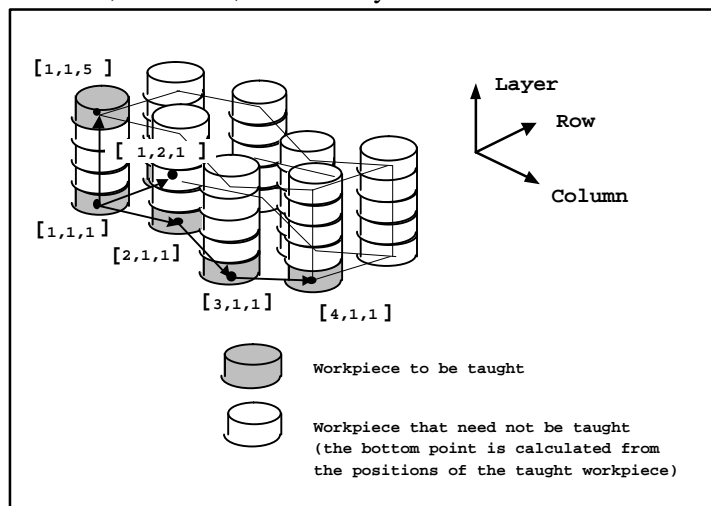
Operation

To palletize or depalletize workpieces, maintaining the same attitude and configuration as that taught, perform the following setup procedure:

- Step**
- 1** On the system variable screen, set system variable \$PALCFG.\$FREE_CFG_EN to TRUE (the initial value is TRUE).
 - 2** On the palletizing initial data screen, set attitude control to INTER for the palletizing direction (one of COLUMN, ROWS, and LAYERS) for which the arrangement method is FREE.

As a result, each workpiece along the specified direction is palletized or depalletized with the same attitude and configuration as that of the reference workpiece used for teaching.

The following is an example of an irregular arrangement of four columns, two rows, and five layers.



Set the palletizing initial data as follows:

```

COLUMN = [ 4 FREE INTER ]
ROWS   = [ 2 LINE FIX   ]
LAYERS = [ 5 LINE FIX 1 ]

```

In this example, FREE and INTER are set for COLUMN. When system variable \$PALCFG.\$FREE_CFG_EN is TRUE, workpieces are palletized or depalletized with the following configurations:

- P[1,1,1] for workpieces in the first column
- P[2,1,1] for workpieces in the second column
- P[3,1,1] for workpieces in the third column
- P[4,1,1] for workpieces in the fourth column

Notes

When using this function, note the following:

- Step 1** FREE and INTER cannot be set for more than one direction of COLUMN, ROWS, and LAYERS (such setting is possible when system variable \$PALCFG.\$FREE_CFG_EN is FALSE, that is, the function is not used).

This is because the configuration for a workpiece that need not be taught (whose bottom point is calculated from the positions of the reference workpieces which are taught) cannot be uniquely determined from such a setting.

When the program contains such a setting, the execution will fail and the following error message is output:

PALT-024 Calculation error occurred

- 2** Create a program so that it is not stopped due to an alarm for an unmatched configuration.

When the configuration at the current position and that at the destination position are different, the robot cannot move between the positions with linear movement (program execution is stopped and an unmatched configuration alarm is output).

The configurations at the approach positions and retreat positions for palletizing are specified in the same way as those for the bottom points. If the first palletizing command specifies linear motion type, an unmatched configuration alarm may be output, depending on the configuration of the robot when the command is executed.

This problem does not occur when the first palletizing command specifies joint motion type.

For example, the following program does not cause an unmatched configuration alarm to be output for palletizing with three approach positions and two retreat positions:

```

:
10: PALLETIZING EX_1
11: J PAL_1[A_3] 100% FINE
12: L PAL_1[A_2] 500mm/sec Cnt50
13: L PAL_1[A_1] 300mm/sec Cnt10
14: L PAL_1[BTM] 100mm/sec FINE
15: OPEN_HAND1
16: L PAL_1[R_1] 300mm/sec Cnt10
17: L PAL_1[R_2] 500mm/sec Cnt50
18: PALLETIZING-END-1
:

```

10.16 COORDINATES OFFSET FUNCTION

The coordinates offset function changes either the tool coordinate system or the user coordinate system for a range of motion instructions in a program for which teaching has been completed. The function then converts the positional data so that the TCP position does not change due to the shift between the original and changed coordinate systems.

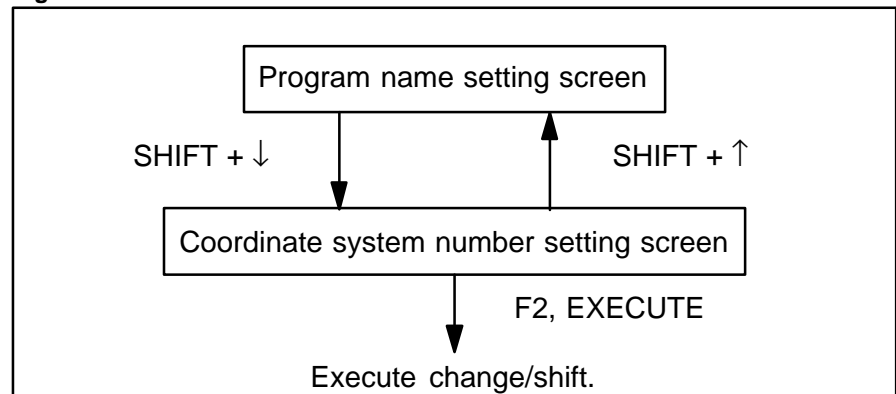
Types of Coordinates Offset

The following two types of coordinates offset are available:

- **TOOL OFFSET** – Changes the **tool** coordinate system number and positional data in a teach pendant program.
- **UFRAME OFFSET** – Changes the **user** coordinate system number for the positional data in a teach pendant program.

Coordinates offset is executed on the TOOL/UFRAME OFFSET screens (UTILITIES, Tool offset). The screens are switched as shown in Figure 10-57.

Figure 10-57. Coordinates Offset Screens



Coordinates Offset

The coordinates offset function performs the following:

- Changes the tool coordinate system number or user coordinate system number for the positional data (Cartesian coordinates) in all or a range of motion instructions in an existing program.
- If the positional data is specified with joint coordinates, converts the data according to the shift resulting from the tool or user coordinate system change.
- Inserts the results of the conversion into a new or existing program.
- Executes the same conversion for other programs, if necessary.

Converting the Positional Data

The positional data is converted according to the following rules:

Position and attitude

- Positional data specified with Cartesian coordinates is converted to Cartesian coordinates. Positional data specified with joint coordinates is converted to joint coordinates.
- If the converted joint coordinates fall outside the operating range, the corresponding positional data is assumed to be untaught. For Cartesian coordinates, the converted position is stored as is.
- The positional data in the position registers is not converted.
- For motion instructions that include the incremental motion option, positional data specified with joint coordinates is assumed to be untaught.

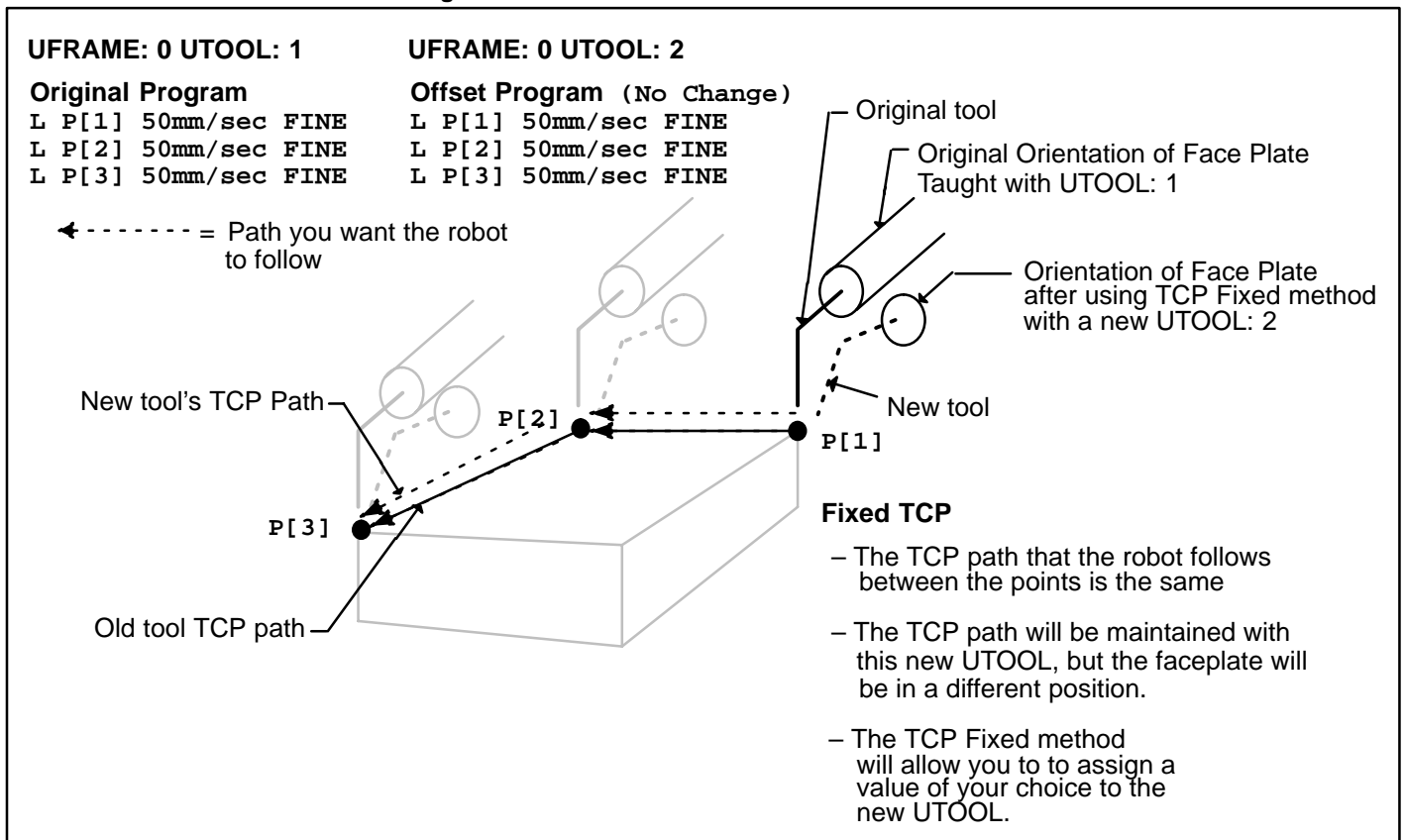
Axis location and rotation speed of positional data specified with Cartesian coordinates

- The same format is used both before and after conversion.
- If the wrist axis is rotated through 180° or more as a result of conversion, the rotation speed for the axis is optimized; a message is then displayed prompting you to select whether to use the optimized rotation speed.

For **UTOOL OFFSET**, you can select either of the following positional data conversion methods:

- **TCP fixed:** This method lets you specify a new TOOL frame number of your choice for use with a new or damaged tool. Programmed positions are not changed. The same TCP path will be maintained with this new UTOOL, but the faceplate position will be different. See Figure 10-58.

Figure 10-58. TCP Fixed Method



- Robot fixed:** This method lets you specify the TOOL frame number to use with the current tool. Programmed positions are automatically adjusted to maintain the desired path. The robot's motion does not change. See Figure 10-59 and Figure 10-60.

Figure 10-59. Robot Fixed Method

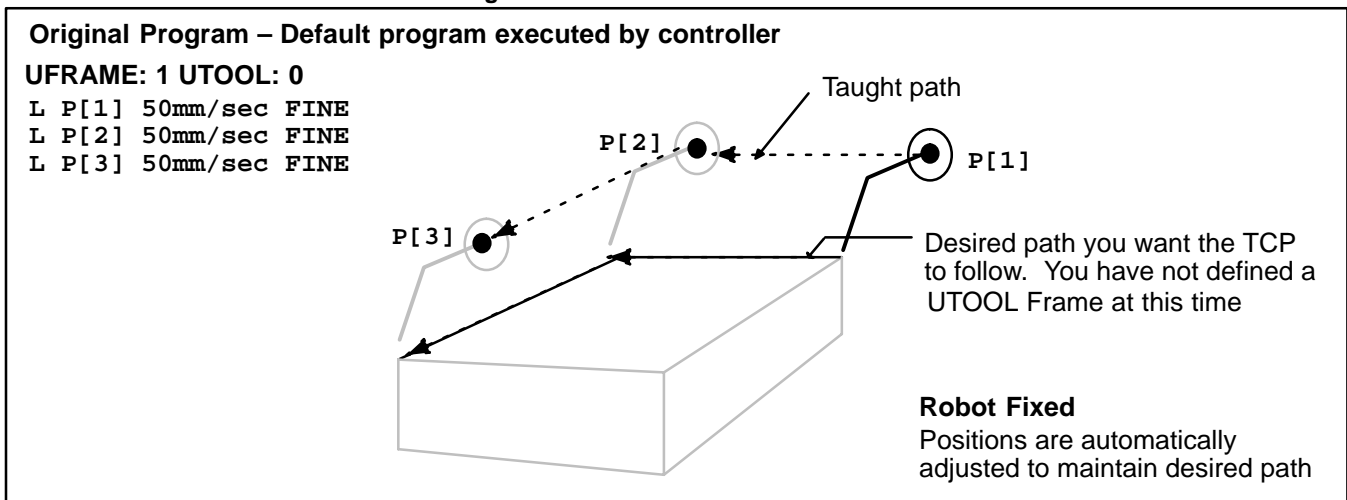
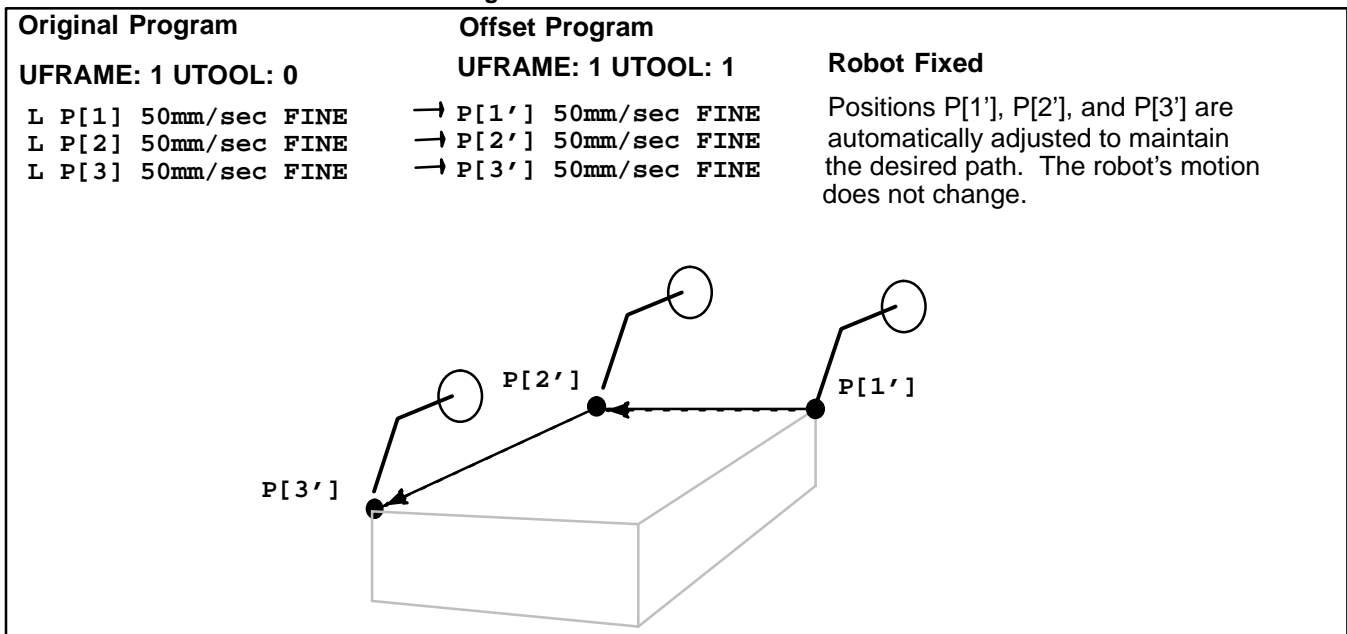


Figure 10-60. Robot Fixed Method



For **UFRAME OFFSET**, you can select whether the positional data is to be converted.

- Convert:** The position data is converted so that the TCP position does not change.
- Not convert:** The position data is not converted even when the coordinate system number is changed.

10.16.1 Tool Frame Offset Function

Table 10-16 lists and describes the items you set to perform the tool frame offset function. Use Procedure 10-15 to perform the tool frame offset function.

Table 10-16. Tool Offset Screen Items

ITEM	DESCRIPTION
Program Name Setting Screen	
Original Program	This is the name of the program that will be offset or shifted.
Range	This specifies the portion of the program that will be offset or shifted: <ul style="list-style-type: none"> • WHOLE – offsets or shifts the entire program • PART – offsets or shifts part of the program
Start line	Not used
End line	Not used
New Program	This is the name of the program that results from offsetting or shifting the Original Program. If you want the resulting offset or shifted program to replace the Original Program, make the New Program name the same as the Original Program name.
Insert line	Used only when all of the following conditions exist: <ul style="list-style-type: none"> • You have not entered a name for the new program, in which case the data conversion will be performed on the currently selected program, or you have entered the name of a program that already exists for the New Program name. • You have selected the Robot Fixed method as the data conversion type. • You have executed the data conversion
Coordinate System Number Setting Screen	
Old UTOOL Number	This is the number of the UTOOL that was used when the positions in the Original Program were recorded.
New UTOOL Number	This is the number of the UTOOL that will be used to offset or shift the program. You must have defined this UTOOL prior to using it. Refer to Section 3.8.1 for information on setting up a tool frame.
Convert Type	This specifies the kind of positional data conversion that will be performed during the offset or shift: <ul style="list-style-type: none"> • TCP fixed - The TCP is maintained during conversion. This means that robot joint positions will change, but Cartesian positions will be fixed. TCP fixed mode can be used, for example, when a damaged hand has been replaced. Specify the tool coordinate system number of the damaged hand for Old UTOOL number and the tool coordinate system number of the replacement hand for New UTOOL number. Then, execute the tool change or shift in TCP fixed mode. The result will be that the TCP of the new tool will move to the originally taught position. • Robot fixed – The robot joint positions are maintained during conversion. Robot fixed mode can be used, for example, when a program has been taught using a tool coordinate system other than that for the mounted hand, after which the tool coordinates have been corrected. Specify the tool coordinate system number used when the program was taught for Old UTOOL number and the corrected tool coordinate system number for New UTOOL number. Then, execute the tool change or shift in Robot fixed mode. The program is modified so that the robot moves according to the corrected tool coordinate system, without changing the resultant robot movement.

Procedure 10-15 Executing a Tool Change or Shift

- Condition**
- A program is to be shifted. See the following screen for an example.
 - The new UTOOL you want to use has been defined.

```

TEST1                                JOINT 30%
                                           1/6
1: J P[1] 100% FINE
      2: J P[2] 70% CNT50
      3: L P[3] 1000cm/min CNT30
      4: L P[4] 500mm/sec FINE
      5: J P[1] 100% FINE

[End]
POINT    SINGLE    DUAL    BACKUP TOUCHUP>

```

- Step**
- 1 Press MENUS.
 - 2 Select UTILITIES.
 - 3 Press F1, [TYPE].
 - 4 Select Tool offset. You will see a screen similar to the following.

```

TOOL OFFSET                            JOINT 10%
Program                                1/6

1  Original Program :    [ TEST1 ]
2  Range:                WHOLE
3  Start line:(not used) ***
4  End   line:(not used) ***
5  New Program :        [ TEST2 ]
6  Insert line:(not used) ***

Use shifted up,down arrows for next page
[TYPE] >

CLEAR >

```

- 5 Move the cursor to the original program, and press ENTER. Use the appropriate function keys to type the program name, and press ENTER.
- 6 Move the cursor to the new program, and press ENTER. Use the appropriate function keys to type the program name, and press ENTER.

- 7 Hold down the SHIFT key and press the down arrow key to display the coordinate system number setting screen. To return to the program name setting screen, hold down the SHIFT key and press the up arrow key.

TOOL OFFSET		JOINT 10%
UTOOL number		1/3
1	Old UTOOL number:	1
2	New UTOOL number:	2
3	Convert type:	TCP fixed
Use shifted up,down arrows for next page		
[TYPE]	EXECUTE	>
CLEAR		>

TCP Fixed Data Conversion Method

- 8 Move the cursor to the Old UTOOL number, type the tool frame number, and press ENTER.
- 9 Move the cursor to the New UTOOL number, type the tool frame number, and press ENTER.
- 10 To convert data using the TCP Fixed method, select 1, TCP Fixed, and press ENTER. A message asking you to confirm the data transformation will appear.
- Press F4, Yes, to execute the transformation.
 - Press F5, No, to cancel the transformation.

Robot Fixed Data Conversion Method

- 11 To convert data using the Robot fixed method, select 2, Robot Fixed, and press ENTER. A message asking you to confirm the data transformation will appear.

NOTE If the "Insert line not set" message is displayed, you are about to perform the data conversion on the original program, or a program that already exists. You will have to enter the number of the line to insert.

- Press F4, Yes, to execute the transformation.
 - Press F5, No, to cancel the transformation.
- 12 Press F2, EXECUTE to execute the data conversion.
- 13 If the rotation speed has changed (been optimized) as a result of conversion, you are prompted whether to use the new rotation speed. See the following screen for an example.

Select P[3]:J5 angle.(deg183)	QUIT>
183° -177° *uninit*	

Select the action you want to take:

- **To use the new, optimized rotation speed**, press F1. The label above F1 indicates that angle that corresponds to the optimized rotation.
- **To use the original rotation speed**, press F2. The label above F2 indicates the angle that corresponds to the original rotation speed.
- **To write the data as untaught data**, press F3, *uninit*.
- **To cancel conversion**, press F5, QUIT.

14 To clear all shift settings, press NEXT, > then press F1, CLEAR.

NOTE After TOOL OFFSET has been executed, the current tool coordinate system number is changed to the newly specified number.

10.16.2 User Frame Offset Function

Table 10–17 lists and describes the items you set to perform the user frame offset function. Use Procedure 10–16 to perform the user frame offset function.

Table 10–17. User Frame Offset Screen Items

ITEM	DESCRIPTION
Program Name Setting Screen	
Original Program	This is the name of the program that will be offset or shifted.
Range	This specifies the portion of the program that will be offset or shifted: <ul style="list-style-type: none"> ● WHOLE – offsets or shifts the entire program ● PART – offsets or shifts a part of the program
Start line	Not used
End line	Not used
New Program	This is the name of the program that results from offsetting or shifting the Original Program. If you want the resulting offset or shifted program to replace the Original Program, make the New Program name the same as the Original Program name.
Insert line	Used only when all of the following conditions exist: <ul style="list-style-type: none"> ● You have not entered a name for the new program, in which case the data conversion will be performed on the original program, or you have entered the name of a program that already exists for the New Program name. ● You have selected the Robot Fixed method as the data conversion type. ● You have executed the data conversion
Coordinate System Number Setting Screen	
Old UTOOL Number	This is the number of the UTOOL that was used when the positions in the Original Program were recorded.
New UTOOL Number	This is the number of the UTOOL that will be used to offset or shift the program. You must have defined this UTOOL prior to using it. Refer to Section 3.8.1 for information on setting up a tool frame.
Convert Position Data	This specifies whether to convert the positional data during the user frame offset: <ul style="list-style-type: none"> ● YES - Converts the positional data so that the TCP does not change during the offset or shift. ● NO - Does not convert the positional data when the coordinate system is changed.

Procedure 10-16 Executing a User Coordinate Change or Shift

- Condition**
- When a program is to be shifted
 - The new UFRAME you want to use has been defined.

```

TEST1                                JOINT 30%
                                      1/6
1: J P[1] 100% FINE
2: J P[2] 70% CNT50
3: L P[3] 1000cm/min CNT30
4: L P[4] 500mm/sec FINE
5: J P[1] 100% FINE
[End]

POINT    SINGLE    DUAL    BACKUP TOUCHUP>

```

- Step**
- 1 Press **MENUS**.
 - 2 Select **UTILITIES**.
 - 3 Press the **F1**, [**TYPE**].
 - 4 Select **Frame offset**. You will see a screen similar to the following (Program Name Setting screen).

```

UFRAME OFFSET                        JOINT 10%
Program                               1/6
1  Original Program : [TEST1]
2  Range:                               WHOLE
3  Start line:(not used)                ***
4  End   line:(not used)                ***
5  New Program : [TEST2]
6  Insert line:(not used)                ***

Use shifted up,down arrows for next page
[TYPE] >

CLEAR >

```

- 5 Move the cursor to the original program, and press **ENTER**. Use the appropriate function keys to type the program name, and press **ENTER**.
- 6 Move the cursor to the new program, and press **ENTER**. Use the appropriate function keys to type the program name, and press **ENTER**.

- 7 Hold down the SHIFT key and press the down arrow key to display the coordinate system number setting screen. To return to the program name setting screen, hold down the SHIFT key and press the up arrow key. See the following screen for an example of the next screen.

UFRAME OFFSET		JOINT 10%
UFRAME number		1/3
1	Old UFRAME number:	<input type="text" value="1"/>
2	New UFRAME number:	<input type="text" value="2"/>
3	Convert Position data (Y/N):	<input type="text" value="YES"/>
Use shifted up,down arrows for next page		
[TYPE]	EXECUTE	>
CLEAR		>

TCP Fixed Data Conversion Method

Robot Fixed Data Conversion Method

- 8 Move the cursor to the Old UFRAME number, type the tool frame number, and press ENTER.
- 9 Move the cursor to the New UFRAME number, type the tool frame number, and press ENTER.
- 10 To convert data using the TCP Fixed method, select 1, TCP Fixed, and press ENTER. A message asking you to confirm the data transformation will appear.
- Press F4, Yes, to execute the transformation.
 - Press F5, No, to cancel the transformation.
- 11 To convert data using the TCP Fixed method, select 2, Robot Fixed, and press ENTER. A message asking you to confirm the data transformation will appear.
- NOTE** If the "Insert line not set" message is displayed, you are about to perform the data conversion on the original program, or a program that already exists. You will have to enter the number of the line to insert.
- Press F4, Yes, to execute the transformation.
 - Press F5, No, to cancel the transformation.
- 12 Press F2, EXECUTE, to execute change or shift.

- 13** If the rotation speed has changed (been optimized) as a result of conversion, you are prompted whether to use the new rotation speed. See the following screen for an example.

```
Select P[3]:J5 angle.(deg183)
183°   -177° *uninit*           QUIT>
```

Select the action you want to take:

- **To use the new, optimized rotation speed**, press F1. The label above F1 indicates that angle that corresponds to the optimized rotation.
 - **To use the original rotation speed**, press F2. The label above F2 indicates the angle that corresponds to the original rotation speed.
 - **To write the data as untaught data**, press F3, *uninit*.
 - **To cancel conversion**, press F5, QUIT.
- 14** To clear all shift settings, press NEXT, > then press F1, CLEAR.

NOTE After FRAME OFFSET has been executed, the current user coordinate system number is changed to the newly specified number.

10.17 TIME BEFORE/AFTER MOTION OPTION INSTRUCTION

Normally, when a teach pendant program is executed, the instruction that follows a motion instruction is not executed until the motion has been completed. The TIME BEFORE/AFTER motion option instruction allows you to specify a teach pendant program that is to be called at a specified time before or after the completion of a motion instruction.

For example, you might specify that a teach pendant program CLS_GRIP is to be called 600 ms before the completion of the move. CLS_GRIP might consist of the instruction DOUT[GRIP]=ON.

This function can reduce external device communication time and improve cycle time. This section contains information on the following:

- Program execution
- Execution timing
- Recording a TIME BEFORE or TIME AFTER instruction
- TIME BEFORE instruction program example
- Programming Hints

10.17.1 Program Execution

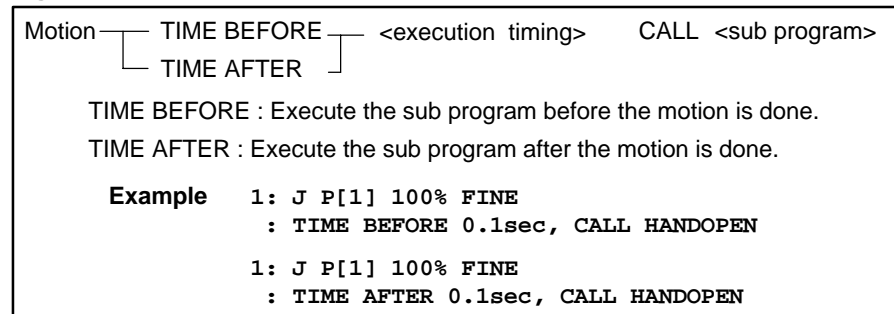
The motion instruction and the sub program (called by the main program) are executed in parallel. Because of this, the execution of the sub program does not affect the robot motion in the main program.

If the robot reaches the destination while the sub program is executing, the robot does not move to the next position until the sub program finishes executing.

You must specify the called program in the TIME BEFORE/AFTER instruction and specify the time when the CALL instruction is to be executed (*execution timing*). If the execution timing is 0 sec, this indicates that the robot has stopped moving. The exact time that robot stops is determined by the termination type (FINE, CNT 100 and so forth).

The called sub program and the execution timing are taught in the motion option instruction. See Figure 10-61.

Figure 10-61. TIME BEFORE / TIME AFTER Motion Option Instructions



Single Step Execution

If you are single stepping through a TIME BEFORE/AFTER instruction, the motion is paused when the sub program is called. The robot moves to the destination position as you single step through the sub program.

Power Fail Recovery

If hot start is enabled and the controller is turned off while the sub program is executing, the sub program will resume execution from the same line the next time the controller is turned on. Because of this, the execution timing of the sub program is different from normal execution.

10.17.2 Execution Timing

Execution timing is the specified time when the CALL instruction is to be executed. Execution timing can be specified as:

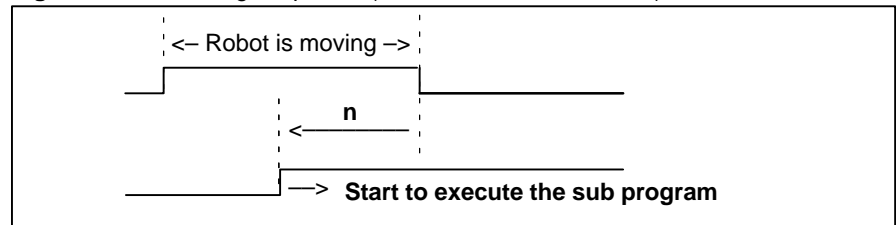
- TIME BEFORE : 0 to 30.0 sec
- TIME AFTER : 0 to 0.5 sec

The execution timing begins counting from the time robot motion is completed. Execution timing is not related to override.

If execution timing is set to 0 sec, the sub program is executed at almost the same time as the statement following the MOVE instruction. When 0 sec is set, the next line of the main program can be executed before the sub program starts to execute. The execution timing acts as follows:

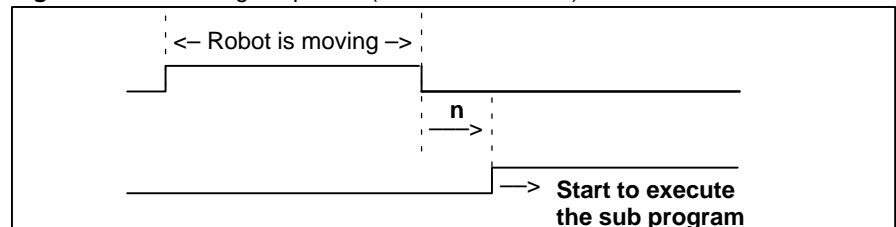
Specify [n sec] in the TIME BEFORE instruction. See Figure 10-62.

Figure 10-62. Timing Sequence (TIME BEFORE instruction)



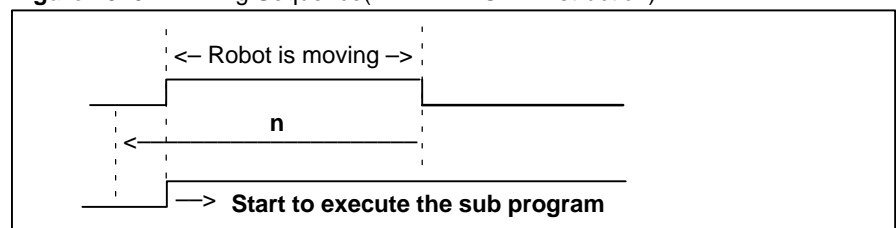
Specify [n sec] in the AFTER instruction. See Figure 10-63.

Figure 10-63. Timing Sequence (AFTER instruction)



The execution timing exceeds the period of the motion. The sub program is executed at the same time motion is started. See Figure 10-64.

Figure 10-64. Timing Sequence (TIME BEFORE instruction)



10.17.3 Recording a TIME BEFORE/AFTER Instruction

Use Procedure 10-17 to record a TIME BEFORE or TIME AFTER Instruction.

Procedure 10-17 Recording a TIME BEFORE or TIME AFTER Instruction

- 1 Move the cursor to the position where you want to add the motion option instruction.

```

PNS0001                                JOINT  10 %
                                         1/2
1: J P[1] 100% FINE ██████
[END]

[CHOICE]

```

- 2 Press F4[CHOICE]. You will see a screen similar to the following.

NOTE To search for the CALL item of a TIME BEFORE or TIME AFTER instruction press F5, [EDCMD], and then select FIND. To replace the TIME BEFORE <-> TIME AFTER, press F5, [EDCMD], and select REPLACE. Then select TIME BEFORE/AFTER. To replace the CALL <program name>, press F5, [EDCMD], and select REPLACE. Then the <program name> can be replaced.

```

Motion Modify                            JOINT  10 %
1                                           5
2                                           6
3 TIME BEFORE                            7
4 TIME AFTER                             8
PNS0001                                JOINT  10 %
                                         1/2
1: J P[1] 100% FINE ██████
[END]

[CHOICE]

```

- 3 Select TIME BEFORE. You will see a screen similar to the following.

```

PNS0001                                JOINT  10 %
                                         1/2
1: J P[1] 100% FINE
:   TIME BEFORE ██████ sec ...
[END]

Enter Value

[CHOICE]

```

- 4 Enter the execution time and press ENTER. For example, enter 2.0 sec.

```

TIME statement                                JOINT  10 %
1 CALL program                                5
2                                                6
3                                                7
4                                                8
PNS0001                                        JOINT  10 %
                                                1/2
1: J P[1] 100% FINE
:   TIME BEFORE 2.0sec 
[END]

Select item                                     [CHOICE]

```

- 5 Select CALL program. A list of available programs will be displayed.

```

PROGRAM list                                JOINT  10 %
1 HANDOPEN                                    5
2 HANDCLOS                                   6
3                                                7
4                                                8
PNS0001                                        JOINT  10 %
                                                1/2
1: J P[1] 100% FINE
:   TIME BEFORE 2.0sec 
[END]

Select item                                     STRINGS

```

- 6 Select the program you want to call with this instruction. For example, HANDOPEN. You will see a screen similar to the following.

```

PNS0001                                        JOINT  10 %
                                                1/2
1: J P[1] 100% FINE
:   TIME BEFORE 2.0sec HANDOPEN 
[END]

[CHOICE]

```


10.17.4 TIME BEFORE Instruction Program Example

Figure 10-65 shows an example main and sub program which illustrate the use of the TIME BEFORE Instruction.

Figure 10-65. Main and Sub Program Examples

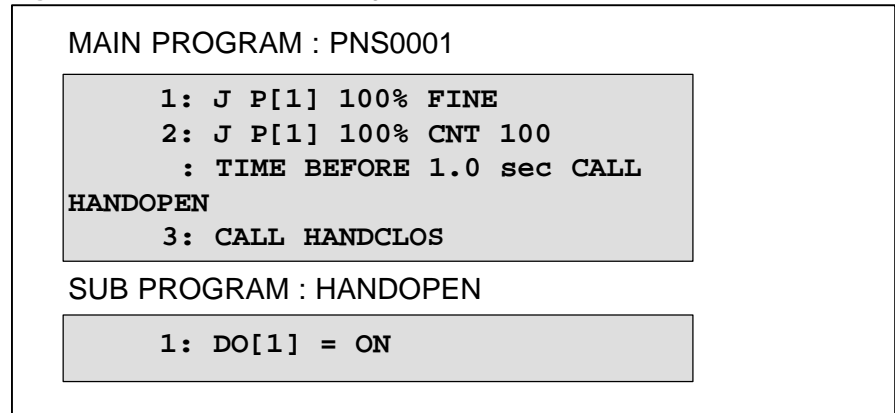
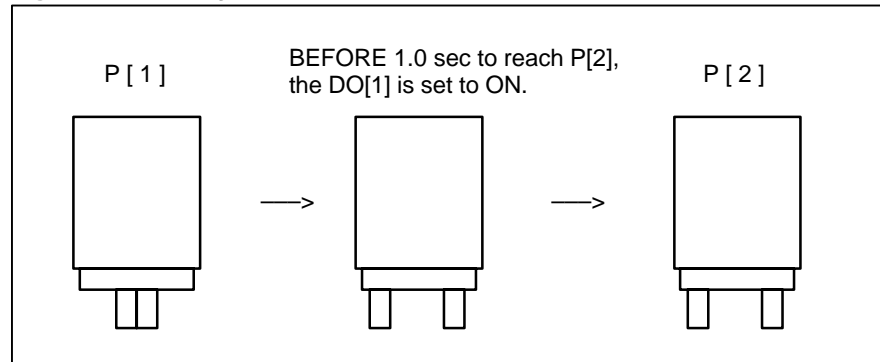


Figure 10-66 shows the execution of the main program, PNS0001.

Figure 10-66. Program example for TIME BEFORE instruction



10.17.5 Programming Hints

The following programming hints apply to the TIME BEFORE or TIME AFTER instruction.

- The sub program called from the TIME BEFORE or TIME AFTER instruction cannot be taught motion instructions. The motion group of the program must be [*,*,*,*,*].
- Until the called program is done executing, the next line cannot be executed.
- There is no limit to the number of lines in a sub program.
- You can use the TIME BEFORE or TIME AFTER instruction in combination with any other motion option instructions, except application instructions such as the SPOT[] instruction or the SKIP instruction.
- Only one TIME BEFORE or TIME AFTER instruction can be used with a single MOVE instruction.
- If you add CNT to a motion statement, the timing when the motion statement is completed is changed by the value of CNT. Even if 0 sec is specified in the TIME BEFORE instruction, the sub program might be executed too soon. You might need to use the TIME AFTER instruction to adjust the execution timing.
- When the TIME BEFORE or TIME AFTER instruction is used in the last line of the program, the sub program might not be called. This is because the execution of the main program is completed before the sub program is called. Therefore, do not teach the TIME BEFORE or TIME AFTER instruction on the last line.

10.18 CONDITION MONITOR FUNCTION

The Condition Monitor Function monitors the condition of an I/O signal, register value, or alarm status, during teach pendant program execution. As soon as the condition is triggered, the specified teach pendant program is executed and interrupts the current program.

A Condition monitor is defined by two or more teach pendant programs:

- A condition (CH) program specifying one or more sets of conditions, such as a port or register value. Each set of conditions contains the name of an action program to be called when the condition is satisfied.
- One or more action programs specifying what is to be done when a condition is satisfied.

For example, you can use the condition monitor function as follows:

If a robot is handling a work piece and drops it, an error message is displayed and the robot pauses. See Figure 10-67 and Figure 10-68.

(CONDITION)<The condition to be monitored>:

[Dropping the work piece] => RDI[2] = OFF

(ACTION)<The program executed when the condition is triggered>:

[Error message] => User alarm[] & [Pause robot]

Figure 10-67. Condition Monitor Function

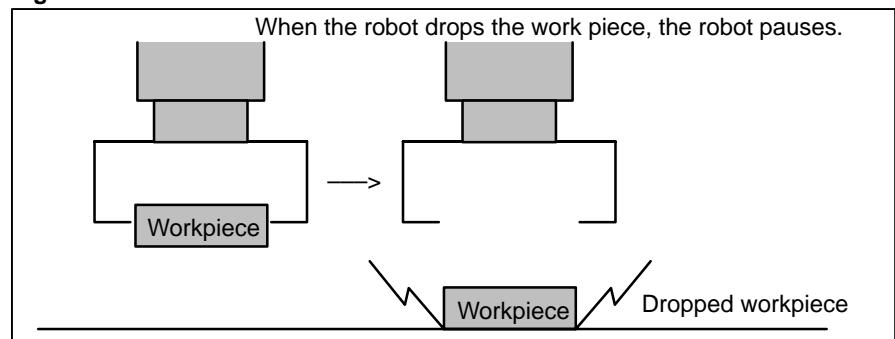


Figure 10-68. Sample, Condition Handler, and Action Programs

```

SAMPLE.TP (to perform handling work)
1: MONITOR WORK_DROP          <--+
   :                           | Monitoring section
   :                           |
9: MONITOR END WORK_DROP      <--+
[END]
WORK_DROP.CH (condition handler program)
1: WHEN RDI[2] = OFF, CALL ROBOT_PAUSE
[END]
ROBOT_PAUSE.TP (action program)
1: SDO[2] = ON                ! Notify the peripheral device
2: R[8] = R[8] + 1            ! Count the number of dropped times
3: User alarm[1]              ! Display alarm and pause the robot
[END]
$UALRM_MSG[1] (system variable)
$UALRM_MSG[1] = 'WORK WAS DROPPED'

```

10.18.1 Monitors

There are two kinds of monitors:

- The **Program monitor** is started by a program monitor instruction and stops monitoring when the program executes a MONITOR END instruction or is aborted.
- The **System monitor** is started and ended using the STATUS System Monitor screen.

Program Monitor

Program monitor is for monitoring conditions in each teach pendant program. This monitor depends on the status of program execution. You start monitoring by using the teach pendant instruction MONITOR. You end monitoring by using the teach pendant instruction MONITOR END or by aborting the program.

SAMPLE.TP

System watches the conditions specified by the ch program.

```

5: MONITOR <ch program name>      ----+
6: J P[4] 100% CNT100              |
7: J P[5] 100% CNT100              |
:                                   |
:                                   |
19: MONITOR END <ch program name>  ----+

```

System Monitor

System monitor does not require a program to be executing for monitoring to take place. Unlike the system monitor, the program monitor only monitors while the program is executing. When the program is aborted, the program monitor terminates. The system monitor is for monitoring the condition of system, like a PLC. You can start and end the system monitor from the condition menu. Unlike program monitor, you cannot start and end the system monitor using teach pendant instructions. A MONITOR instruction in the action program of a system monitor can be used to restart the system monitor.

You can use the system variable \$TPP_MON.\$global_mt to select a mode type at cold start.

- Type1 – If the monitor is executing before power off, the system deletes the monitor at cold start.
- Type2 – If the monitor is executing before power off, the system starts monitoring at cold start automatically.

NOTE You cannot use TYPE1 and TYPE2 together.

NOTE You can use system monitor and program monitor concurrently.

Changing the Monitor Type

You can change the type of monitor as follows:

\$TPP_MON.\$local_mt = 1 – Program monitor TYPE1 (default)

\$TPP_MON.\$local_mt = 2 – Program monitor TYPE2

\$TPP_MON.\$global_mt = 0 – No use system monitor(default)

\$TPP_MON.\$global_mt = 1 – System monitor TYPE1

\$TPP_MON.\$global_mt = 2 – System monitor TYPE2

**10.18.2
Monitor State**

The following table shows the state of the monitor by each operations.

@ : Start the monitor

o : Restart the monitor if the monitor was executing at paused

% : Pause the monitor (It can restart)

x : Cancel the monitor (It can not restart)

– : It does not change the state of monitor

Table 10-18. State of Condition Monitoring

OPERATION	Program monitor		System monitor	
	TYPE 1	TYPE 2	TYPE 1	TYPE 2
MONITOR (Teach pendant instruction)	@	@	–	–
START (Function key at condition menu)	o	o	@/o	@/o
Program is PAUSED	%	–	–	–
Program is ABORTED	x	x	–	–
MONITOR END (Teach pendant instruction)	x	x	–	–
PAUSE (Function key at condition menu)	%	%	%	%
END (Function key at condition menu)	x	x	x	x
RESTART (Function key at condition menu)	o	o	–	–
HOT START(Power down at teach pendant program execution)	%	–	–	–
HOT START(Power down at teach pendant program stop)	–	–	–	–
COLD START	x	x	x	–
CTRL START	x	x	x	x

10.18.3 Monitor Instructions

A program monitor is executed using the following two instructions:

- **MONITOR** <ch program>
Start monitoring the conditions taught in the <ch program>.
- **MONITOR END** <ch program>
Stop monitoring the conditions taught in the <ch program>.

You can use the system variable \$TPP_MON.\$local_mt to change monitoring modes while a program is PAUSED.

- Type1 – Stop monitoring when the program is PAUSED.
- Type2 – Keeps on monitoring even when the program is PAUSED.

NOTE You cannot use Type1 and Type2 together.

10.18.4 Condition Handler Program

You can teach the condition to the program whose sub type is Cond. When editing the condition handler program, only the WHEN instruction is available.

```
WHEN <condition> CALL <program>
```

In a condition handler program, you can teach multiple WHEN instructions as follows.

```
1: WHEN <cond1> CALL <program1>
2: WHEN <cond2> CALL <program2>
3: WHEN <cond3> CALL <program3>
```

You can connect the multiple conditions using AND/OR as follows.

```
1: WHEN <cond1> AND <cond2> CALL <program1>
2: WHEN <cond1> OR <cond2> OR <cond3> CALL
<program2>
```

NOTE You cannot use both AND and OR in the same WHEN instruction.

10.18.5 Conditions

Figure 10-69 through Figure 10-71 show the conditions that can be monitored.

Figure 10-69. Condition for Register, System Variable, and I/O Parameters

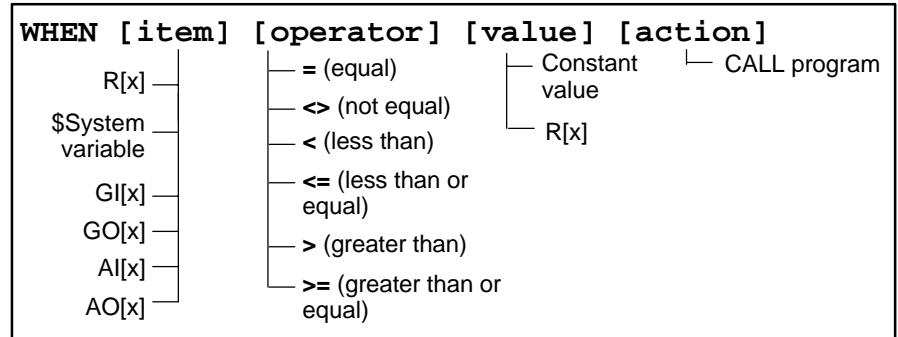


Figure 10-70. Condition2 for I/O

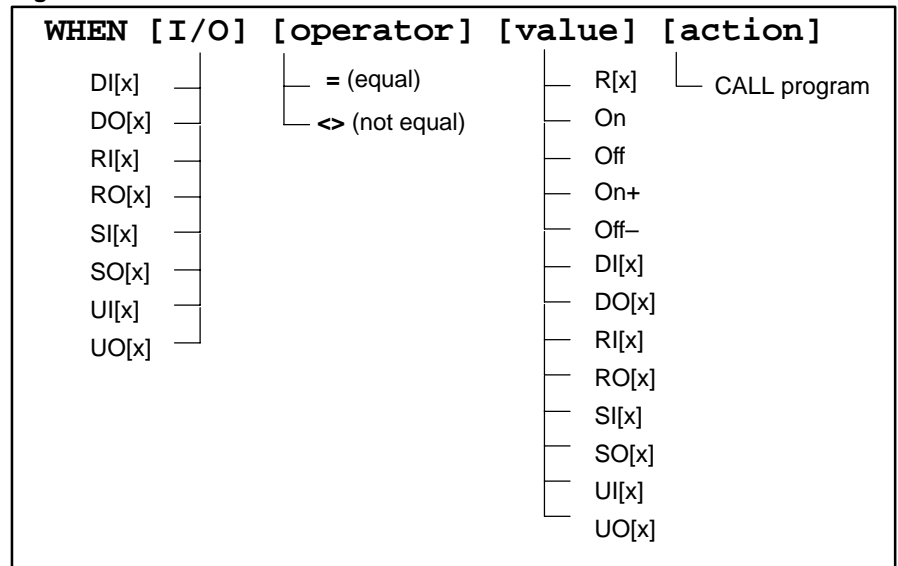
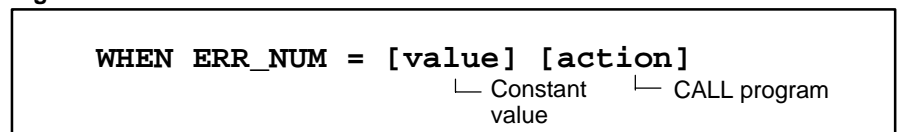


Figure 10-71. Condition for Error status



ERR_NUM = aaabbb

- aaa : Error facility code (decimal); Refer to Section A.1.1.
- bbb : Error number (decimal)

Example: WHEN ERR_NUM=11006, CALL PROG_A

This means “SRVO-006 Hand broken” error because the SRVO facility code is 11.

If 0 is specified as error number “aaabbb”, whenever any error occurs, the condition is satisfied.

10.18.6 Condition Menu

This menu has the following functions:

- Program monitor
 - Displays the status of program monitor
 - Restarts the program monitor
 - Pauses the program monitor
 - Ends the program monitor
- System monitor
 - Displays the status of system monitor
 - Starts or restarts the system monitor
 - Ends the system monitor

To select a condition menu

1. Press STATUS.
2. Press F1, [TYPE].
3. Select Condition.

Program Monitor Menu

You can see the following menu for information of program condition. This menu lists the running or paused program condition only. See Figure 10-72. Refer to Table 10-19 for a description of the items on the Program Monitor menu.

NOTE The Program Monitor menu does not display conditions that have not been started.

Figure 10-72. Program Monitor Menu

Program monitor				JOINT	10 %
				1 / 3	
	CH Prog.	Status	Program		
1	WORK_DRP	Running	SAMPLE		
2	HAND_CHK	Paused	SAMPLE		
3	HAND_CHK	Paused	SAMPLE2		
[TYPE] SYSTEM RESTART PAUSE END					

Table 10-19. Program Monitor Menu Items

ITEM	DESCRIPTION
CH Prog.	This is the name of active condition handler program.
Status	This is the status of condition. <ul style="list-style-type: none"> – Running : The monitoring of this condition is enabled. – Paused : The monitoring of this condition is disabled.
Program	This is the name of the program that starts the condition. If the sub program starts the monitor, the main program name is displayed.

Table 10-19. (Cont'd) Program Monitor Menu Items

ITEM	DESCRIPTION
SYSTEM	This displays the System Monitor screen. NOTE If \$TTP_MON.\$global_mt equal to 0, then this function key does not work and display the message "System monitor is not available".
RESTART	This restarts the paused condition.
PAUSE	This pauses the program condition.
END	This ends this condition. The status is set to canceled and the condition stopped.

System Monitor Menu

You can see the information for system condition and operate the system condition. See Figure 10-73. Table 10-20 lists and describes the items on the System Monitor menu.

Figure 10-73. System Monitor Menu

```

System monitor                               JOINT 10 %
-----
CH Prog.  Status
1  WORK_DRP  Running
2  HAND_CHK

[ TYPE ]  PROGRAM                START  END

```

Table 10-20. System Monitor Menu Items

ITEM	DESCRIPTION
CH Prog.	This lists the condition programs.
Status	This is the status of the condition. <ul style="list-style-type: none"> – Running : The monitor of this condition is enabled. – Paused : The monitor of this condition is disabled. – (blank) : This condition has not yet started or has ended.
PROGRAM	This displays the program condition screen.
START	This starts or restarts system conditions.
END	This ends this condition. The status is canceled and the display is cleared.

You can change the type of system monitors by changing the system variable \$TPP_MON.\$GLOBAL_MT as follows. You can only change this system variable in system variable menu at CTRL START.

- \$TPP_MON.\$GLOBAL_MT = 0 – No use system monitor(default)
- \$TPP_MON.\$GLOBAL_MT = 1 – System monitor TYPE1
(see “The state of monitor”)
- \$TPP_MON.\$GLOBAL_MT = 2 – System monitor TYPE2
(see “The state of monitor”)
You can start and end the monitor at condition menu in STATUS menu. (see “Condition menu”)

10.18.7 Restrictions

The multiple conditions taught in the condition handler program are monitored at the same time.

```
1: WHEN <cond1> CALL <program1>
2: WHEN <cond2> CALL <program2>
3: WHEN <cond3> CALL <program3>
```

When the next conditions begin to be monitored before the last conditions are stopped, then both conditions are monitored at the same time.

A program monitor is canceled in the following cases:

- One of the conditions is triggered.
- Execute the “MONITOR END” teach pendant instruction.
- The program is aborted.
- The END function key, on the program monitor screen, is pressed.

NOTE In the program monitor TYPE1 (\$TPP_MON.\$local_mt=1), when the program is paused, the program monitor is paused. The program monitor is restarted by the program restart.

A system monitor is cancelled in the following ways:

- One of the conditions is triggered.
- A cold start is executed and \$TPP_MON.\$global_mt = 1.
- The END function key, on the program monitor screen, is pressed.

A program or system monitor can be restarted after it triggers, by having the action program it calls execute a MONITOR teach pendant instruction.

The maximum number of conditions connected with AND/OR operator is 5. The total number of monitors is limited to 50.

```

                                Max 5 conditions
                                +-----+
                                |         |
                                |         |
WHEN <cond1> AND <cond2> ... AND <cond5> ---+
WHEN <cond1> OR <condm> ... OR <condp>      |Max
                                :           :      |50
                                :           :      |
WHEN <conds> AND <condt> ... AND <condw> ---+

```

You cannot execute motion statements in the action program when the robot is moving.

You cannot edit the active ch program.

The group mask of the action program for a system monitor must be [*,*,*,*].

You can specify the group mask of the action program for the program monitor. However, the action program cannot move the robot when the robot is moving.

When the condition is triggered, the monitor state becomes “end”. If you want to continue monitoring, you should teach a “MONITOR” instruction in the action program. In this time, the action program should disable the condition. See the following example.

```

MAIN.TP
1: MONITOR MON1
:
9: MONITOR END MON1

MON1.Cond
1: WHEN R[1]=1 CALL ACT1

ACT1.TP
1: R[1]=0 <--- disable the condition
2:
3: ( action )
4:
5: MONITOR MON1 <--- restart monitor

```

You cannot execute to the ch program directly.

Use Procedure 10-18 to create a condition handler program.

Use Procedure 10-19 to create an action program.

Use Procedure 10-20 for an example of creating a condition handler program.

Use Procedure 10-21 to start a condition handler program from a teach pendant program.

Procedure 10-18 Creating a Condition Handler Program

- 1 Press SELECT.
- 2 Press F2, CREATE.
- 3 Enter the program name (CH program name).
- 4 To display program header information,
 - a Press F2, DETAIL.
 - b Move the cursor to the sub type and press F4, [CHOICE]. You will see a screen similar to the following.

Sub Type		JOINT 10%
1	None	
2	Macro	
3	Cond	
4		
Program Detail		
1	Program Name	[CHK_CELL]
2	Sub Type:	[]
		[CHOICE]

- c Select cond.

NOTE If you set the sub type to cond, the system sets the group mask to [*,*,*,*] automatically. You cannot change the group mask.

- 5 When you have finished entering program information, press F2, END.
- 6 Press F1,[INST]. You will see a list of WHEN instructions. See the following screen for an example.

WHEN statement	
1 WHEN ...=...	5 WHEN ...>...
2 WHEN ...<>...	6 WHEN ...>=...
3 WHEN ...<...	
4 WHEN ...<=...	

Execution Sequence

The following is the sample of the program monitor.
 Condition: DI[1] turn on
 Action: DO[1] turn on

Procedure 10-19 Creating an ACTION Program

- 1 Press SELECT.
- 2 Press F2,CREATE.
- 3 Enter the program name (ex. ACT)
- 4 Display program header information to change the group mask.
 - a Press F2, DETAIL.
 - b Change the group mask to [*,*,*,*,*]
- 5 Teach the following instruction.
 ACT.TP(**group mask** = [*,*,*,*,*])
 1: DO[1]=ON

NOTE The group mask of the action program for the system monitor must be set to [*,*,*,*,*].

Procedure 10-20 Creating a Condition Handler Program (Example)

- 1 Press SELECT.
- 2 Press F2,CREATE.
- 3 Enter the program name (ex. COND1)
- 4 To display program header information, press F2, DETAIL.
 - a Move the cursor to sub type and press F4, [CHOICE].
 - b Select cond.
 - c Press F2, END.
- 5 Teach the instruction.
 COND1.TP(sub type = COND, group mask = [*,*,*,*,*])
 1: **WHEN DI[1]=ON+,CALL ACT**

COND1	JOINT 10 %
	1/1
[End]	
[INST]	[EDCMD]

Procedure 10-21 Starting a Condition Handler Program from a Teach Pendant Program

- 1 Press SELECT.
- 2 Press F2,CREATE.
- 3 Enter the program name (ex. MAIN.TP)
- 4 Teach the instruction.

MAIN.TP

1: MONITOR COND1
 2: WAIT 10.00(sec)
 3: MONITOR END COND1

MAIN	JOINT 10 %
[End]	1/1
[INST]	[EDCMD]

- 5 Press F1 [INST]. You will see a screen similar to the following.

Instruction	JOINT 10 %
1 Registers	5 JMP LBL
2 I O	6 CALL
3 IF SELECT	7
4 WAIT	8 MONITOR/MON. END
MAIN	1/1
[End]	
Select item	[CHOICE]

- 6 Select MONITOR/MON. END. You will see a screen similar to the following.

```

MONITOR statement          JOINT 10 %
1 MONITOR                  5
2 MONITOR END              6
3                           7
4                           8
MAIN
                               1/1
[End]
Select item
                               [CHOICE]

```

- 7 Select MONITOR and display the list of ch programs.

```

Cond. PROGRAM list        JOINT 10 %
1 COND1                   5
2                           6
3                           7
4                           8
MAIN
                               1/2
  1: MONITOR
[End]
Select item
                               [CHOICE]

```

- 8 Teach the following program.

```

MAIN                      JOINT 10 %
                               1/3
  1: MONITOR COND1
  2: WAIT 10.00(sec)
  3: MONITOR END COND1
[End]
Select item
                               [CHOICE]

```

- 9 Start the "MAIN" program.
- 10 If you turn on the DI[1], the DO[1] will turn on while the program executes the second line.

10.19 COLLISION GUARD (OPTION)

The Collision Guard option provides a highly sensitive method to detect that the robot has collided with an object and stop the robot immediately. This helps to minimize the potential for damage to the end-of-arm tooling and robot.

Collision Guard can be used in any application but is especially useful in applications in which a large amount of force is applied, such as stud or pedestal welding. It can also be used in applications in which the robot payload changes, such as in handling applications.

Collision Guard also helps to prevent damage during teaching.

The ability to disable the option selectively allows you to use it when some disturbances are applied to the robot, as long as you can predict in your program when these disturbances will occur.

Collision Guard is in effect both during jogging motion and programmed motion whenever it is enabled.

There are several ways to configure and adjust Collision Guard:

- The **Collision Guard Setup screen** allows you to enable and disable Collision Guard globally, for both programmed motion and jogging motion.

In addition, you can use this screen to adjust the sensitivity of collision detection for programmed motion.

Collision Guard automatically uses more sensitive limits for jogging motion. These limits can not be adjusted. You can still disable Collision Guard for jogging motion, using the Collision Guard Setup screen.

- Within a **teach pendant program**, you can disable Collision Guard locally through the use of special teach pendant instructions, COL DETECT OFF and COL DETECT ON.

For Collision Guard to operate properly, you must set payload information correctly. Refer to Section 3.20.

NOTE In order to decrease the force of collision, Collision Guard allows the robot axes to sag away from the collision for 200 milliseconds after detecting a collision. When this happens, vertical robot axes might fall slightly after detecting a collision, due to the effect of gravity.

10.19.1 Limitation

You **cannot use** Collision Guard when the robot brakes are on. Collision detection is disabled automatically when the softfloat function is enabled.

10.19.2 Falsely Detected Collisions

Collision Guard might detect a false collision when a collision has not occurred in the following cases:

- Payload information has not been set correctly.
- The ACC motion option has been used, causing jerky robot motion.
- Not enough voltage has been supplied to the controller.
- The payload is larger than the maximum payload for the robot, or the inertia of the payload is too large.
- Very high speed rotations of wrist joints occur with improperly set payload parameters.
- Jerky reverse motion (P[1]→P[2]→P[1]) occurs.
- Linear motion occurs near singular point where axes revolve in high speed.

10.19.3 Collision Guard Adjust Macro Program

You can use the Collision Guard Adjust macro program, CG_ADJUST, to set the Collision Guard sensitivity during program execution. You must use the CG_ADJUST macro program with the Sensitivity Macro Register.

The Sensitivity Macro Register is a register that contains the Collision Guard sensitivity value. The sensitivity value is a value from 1% to 200%, where 1 is least sensitive and 200 is most sensitive.

Adjusting Collision Guard Sensitivity within a Program

To adjust Collision Guard sensitivity within a program, do the following:

1. Add the CG_ADJUST macro program to the macro table. (Section 3.10)
2. Specify the Sensitivity Macro Register number on the COL GUARD SETUP screen. (Procedure 10-22)
3. Add the following instructions to your program, each time you want to set the Collision Guard sensitivity:
 - A **register assignment instruction** – to assign the sensitivity value you want to the Sensitivity Macro Register you specified on the COL GUARD SETUP screen.
 - A **macro instruction**, CG_ADJUST, to run the CG_ADJUST macro program.

See Figure 10-74.

Figure 10-74. Collision Guard Adjust Macro Program

•		
7: R[7]=120 ←		Assigns a Collision Guard sensitivity value of 120% to R[7], the Sensitivity Macro Register specified on the COL GUARD SETUP screen.
8: CG_ADJUST ←		Collision Guard Adjust macro program will set the sensitivity to the value specified in R[7], the Sensitivity Macro Register.
•		

FANUC Robotics recommends using the CG_ADJUST macro program only after motion instructions that use the FINE termination type.

**WARNING**

When the CG_ADJST program is executed, if the robot is in motion, it will come to a stop momentarily while it executes CG_ADJST. If the CNT termination type is being used for the motion, the robot will stop at the destination position before proceeding to the next position, instead of moving to that position with continuous termination type.

Include the CG_ADJST program after motion instructions that use FINE termination type. Otherwise, personnel could be injured and equipment damaged.

10.19.4 Setup

Before you can use Collision Guard, you must set it up. Setup includes

- Enabling and disabling Collision Guard
- Setting the Collision Guard Sensitivity
- Specifying a register in which to set and store the sensitivity value for the Collision Guard macro program, if desired

See Table 10-21 for the Collision Guard items you can set up.

Table 10-21. Collision Guard Setup Items

ITEM	DESCRIPTION
Collision Guard Status default: ENABLED	Collision Guard Status specifies whether Collision Guard is on or off: <ul style="list-style-type: none"> • ENABLED indicates that Collision Guard is ON in all cases (programmed and jogging motion), unless it is turned OFF using the COL DETECT OFF instruction in a teach pendant program. • DISABLED indicates that Collision Guard is OFF in all cases (programmed and jogging motion). When Collision Guard Status is set to DISABLED, if you use a COL DETECT ON instruction in a teach pendant program, nothing will happen, Collision Guard will not be ENABLED.
Sensitivity default: 100 % minimum: 1 % maximum: 200 %	Sensitivity allows you to set the level of sensitivity for Collision Guard: <ul style="list-style-type: none"> • 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 Macro Register	Sensitivity Macro Register allows you to specify the register that can be used with the Collision Guard Adjust macro program (CG_ADJST) to adjust the sensitivity of Collision Guard within a program. Refer to Section 10.19.3. A register number of 0 indicates that the register is not used.

Use Procedure 10-22 to set up Collision Guard.

Procedure 10-22 Setting Up Collision Guard

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select COL GUARD. You will see a screen similar to the following.

```
COL GUARD SETUP                                WORLD 10 %
                                                1/3
1 Collision Guard status:                      ENABLED
2 Sensitivity:                                  100%
3 Sensitivity Macro Reg.:                      R[ 7]
[ TYPE ] HELP                                ENABLED DISABLED
```

- 5 **To display help information**, press F2, HELP. When you are finished displaying help, press PREV.
- 6 Move the cursor to the items you want to set and set them as desired.

10.19.5 Programmed Motion

COL DETECT ON COL DETECT OFF

COL DETECT OFF

COL DETECT ON

You can use the following teach pendant instructions to control Collision Guard during programmed motion:

- COL DETECT ON, COL DETECT OFF
- PAYLOAD [GPx:y]

By default, Collision Guard is enabled.

- To **disable** Collision Guard, include the COL DETECT OFF instruction in a teach pendant program.
- To **enable** Collision Guard that has been disabled previously, include the COL DETECT ON instruction in a teach pendant program. Since Collision Guard is always enabled by default, you need to use the COL DETECT ON instruction only if you have previously used the COL DETECT OFF instruction.

See Figure 10-75 for an example of how to use these instructions in a teach pendant program.

Figure 10-75. Example of Enabling and Disabling Collision Guard in a Teach Pendant Program

```

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[5] 50% FINE

```

PAYLOAD [GPx:y]

Collision Guard requires the proper setting of payload information. If the payload changes during your application, you must use the PAYLOAD[x] instruction to select the appropriate payload schedule. Refer to Section 6.23 for details on the PAYLOAD[x] instruction.

Before you use a PAYLOAD[GPx:y] instruction, you must make sure you have set up the payload schedule that corresponds to the one you specify. Refer to Section 3.20 for information on setting up payloads.

10.20 ERROR RECOVERY (OPTION)

A robot program can stop executing during production as a result of various alarms. For example, a welding robot stops moving and welding if a HOLD or EMERGENCY STOP input is detected. Another alarm example is the “ARC-013 Arc Start failed” alarm. In some cases, you might want to clean the torch and cut the wire before resuming the paused welding program. You can use *Error Recovery* to perform these operations automatically and eliminate the time required to jog the robot to and from a manual repair station.

This section is organized as follows:

- Overview – Resume Program, Maintenance Program
- Features
- Limitations
- I/O interface
- Setup
 - Alarm code monitoring
 - Digital input alarms
- Programming
- Testing
- Manual function
- I/O timing sequence

10.20.1 Overview

Error Recovery can execute two kinds of recovery programs: Resume Programs and Maintenance Programs. The primary difference is when and where the recovery programs are executed:

- **Resume Programs** are executed from the point of the error.
- **Maintenance Programs** are executed after exiting the original program.

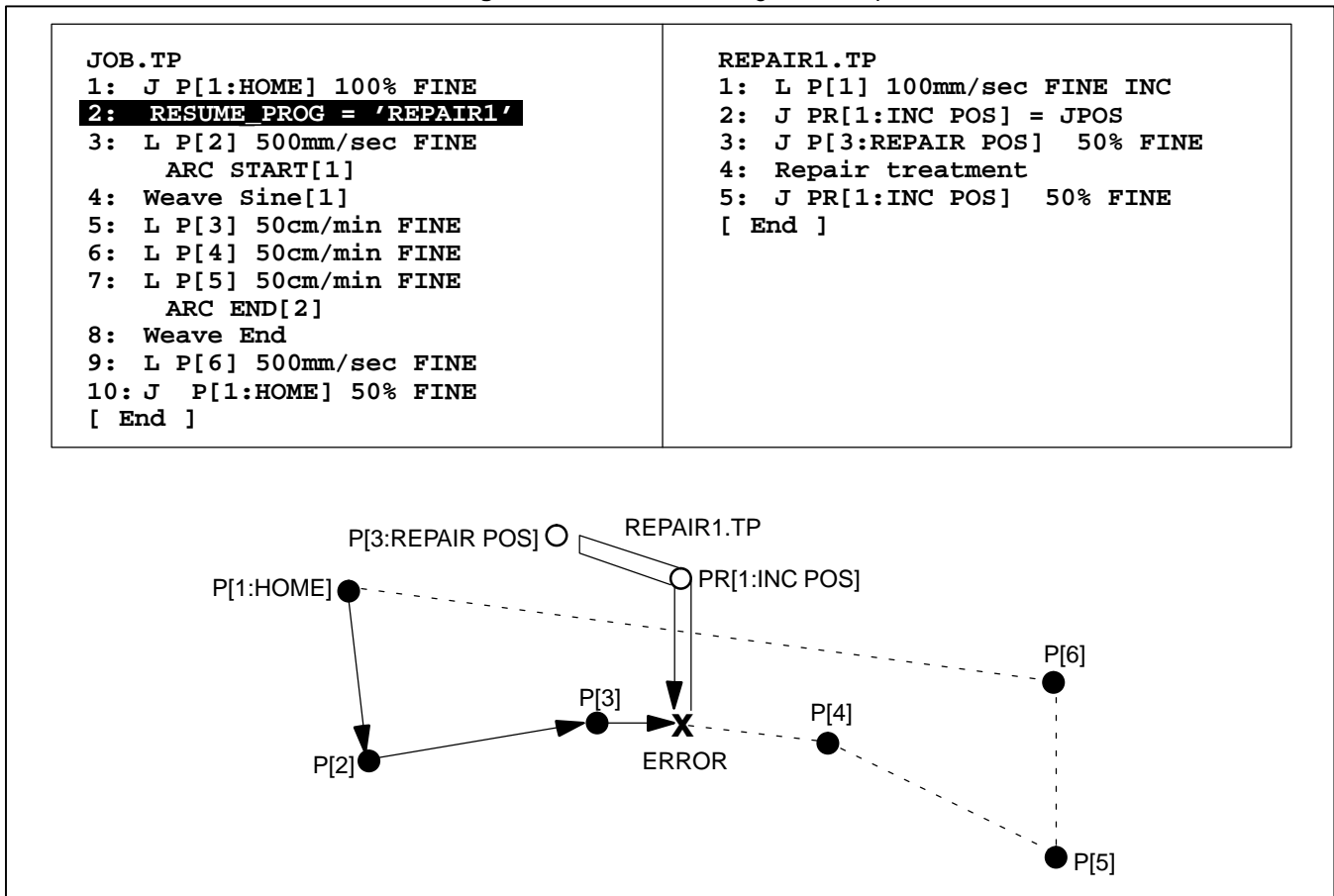
The following two examples illustrate these differences. In both cases, the user program JOB.TP encounters an error after the robot passes position P3.

Resume Program

See Figure 10-76 for an example of a Resume Program:

1. JOB.TP defines REPAIR1.TP as a resume program.
2. An error occurs between positions P3 and P4.
3. REPAIR1.TP is executed from the point of the error.
4. When the REPAIR1.TP completes, JOB.TP is resumed.

Figure 10-76. Resume Program Example



Use a Resume Program when you can define a clear path for the tool and robot from any error position to the recorded positions in the Resume Program. An incremental move away from the error position is often a good first step in a resume program.

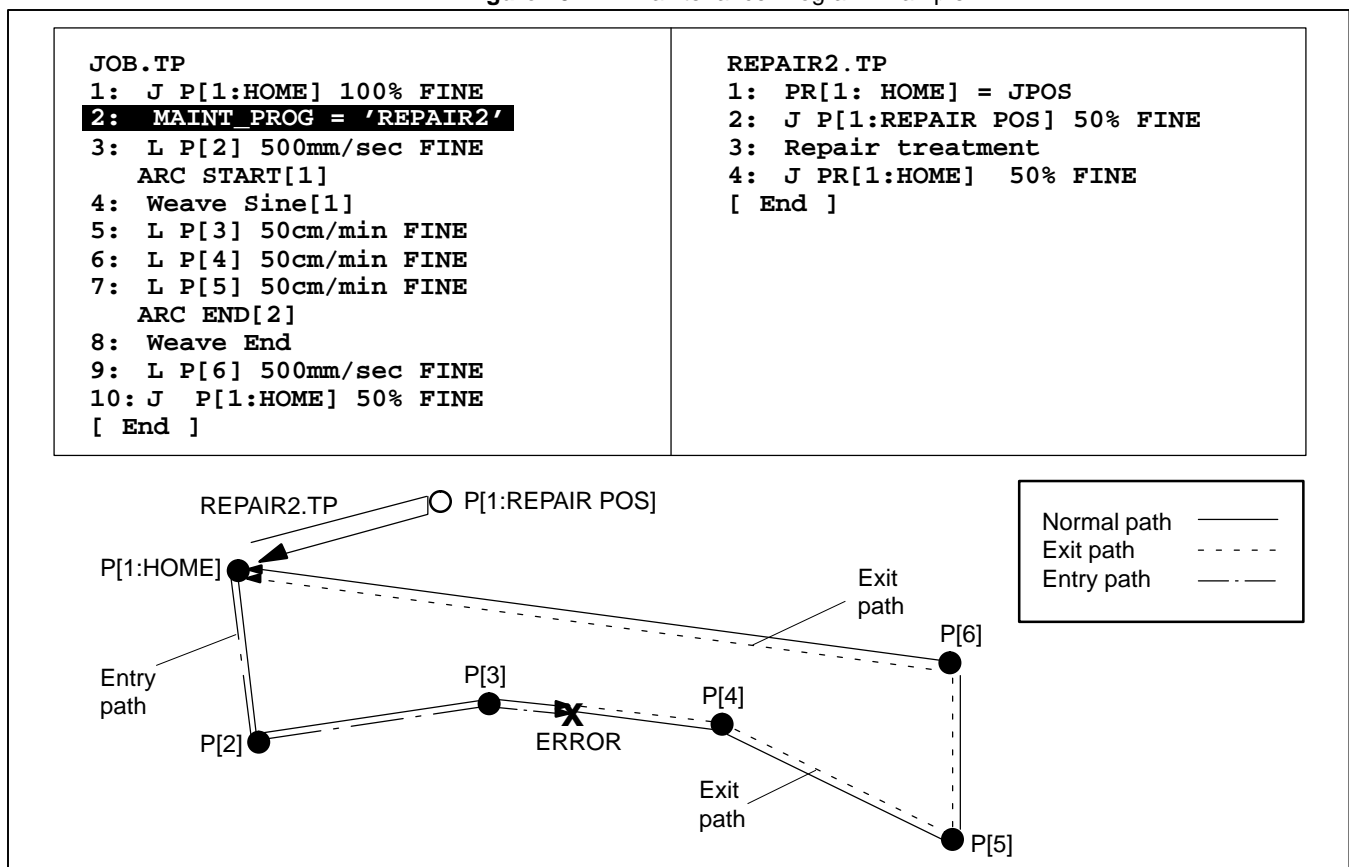
Do not use a Resume Program if your teach pendant program and workpiece configuration do not allow for simple moves away from the recorded positions without colliding with an object. In this case, you can try using a Maintenance Program, which is described in the next section.

Maintenance Program

See Figure 10-77 for an example of a Maintenance Program:

1. JOB.TP defines REPAIR2.TP as a maintenance program.
2. An error occurs between positions P3 and P4.
3. The paused program is “exited” along the original programmed path with the application process (such as welding) turned OFF. This is the **exit path**, which is shown in Figure 10-77 as the **dashed line**.
4. REPAIR2.TP is executed from the HOME position, P1.
5. JOB.TP is re-executed from the beginning of the program to the point of the error with the application process turned OFF. This is the **entry path**, which is shown in Figure 10-77 as the **dot-and-dashed line**.
6. When the point at which the error occurred is reached, the JOB.TP is resumed with the application process turned ON.

Figure 10-77. Maintenance Program Example



Use a Maintenance Program when you cannot define a clear path for the tool and robot from any error position to the recorded positions in the resume program, or any other time.

10.20.2 Features

Table 10–22 summarizes the features available in the Error Recovery option.

Table 10–22. Error Recovery Features

FEATURE	DESCRIPTION
Alarm Code Monitoring	Error Recovery can execute for all alarms or for only a set of specific alarms. Refer to Procedure 10–24 .
Alarm Recovery Screen	SpotTool+ products include an Alarm Recovery screen. This screen allows you to make recovery choices from the teach pendant.
Automatic Start	<p>Automatic Start permits Error Recovery to execute the recovery sequence without waiting for the START input. The fault output is also suppressed. Refer to the I/O timing diagrams in Section 10.20.9.</p> <p>Typically, when an alarm is defined using the Alarm Code Monitoring feature and an alarm occurs, the program is paused with the output of a fault signal. After the first START signal input is received, the Resume Program is executed. After the completion of the Resume Program execution, a second START signal input is received and the paused original program is resumed.</p> <p>If the Automatic Start feature is enabled, when the defined alarm occurs, the Resume Program is executed automatically without the FAULT signal output and without stopping the robot. After the completion of the Resume Program execution, the original program is resumed automatically. Therefore, if the Automatic Start feature is enabled, you do not need to input two START signals.</p>
Resume Programs	Resume Programs allow user-programmed error recovery at the point of the error.
Maintenance Programs	Maintenance Programs allow user-programmed error recovery after exiting the original program.
Program Exit and Entry	Error Recovery automatically exits and enters a user program when using a Maintenance Program.
Teach Pendant Program Instructions	You use teach pendant instructions to define the names of the resume programs and maintenance programs in your teach pendant program.
Error Recovery Status DO	You can define a digital output signal to allow an external control device (such as a PLC) to monitor the recovery process. Refer to the I/O timing diagrams in Section 10.20.9.
Error Recovery Approval DI	You can define a digital input signal to allow an external control device (such as a PLC) to approve or disapprove the execution of the recovery program. Refer to the I/O timing diagrams in Section 10.20.9.
Process Disable	Error Recovery disables welding and weaving during resume program execution, Exit and Entry paths, and maintenance program execution.
Dry Run Speeds	You can change the speed of recovery motions by using dry run speeds during Exit and Entry moves.
Test Mode	You can test Error Recovery execution from the teach pendant using the MANUAL FUNCTIONS screen.

NOTE Use Error Recovery only when the teach pendant is disabled. When the teach pendant is enabled, Error Recovery programs can be executed only from the Manual Function screen. Refer to Section 10.20.8.

10.20.3 Limitations

Error Recovery is DISABLED when the following functions are installed:

- Line tracking
- Soft float
- Continuous turn
- Coordinated motion

In addition, Error Recovery has the following limitations:

- Single step execution is disabled during Resume Program execution. Single step mode is available only for execution of the original program and the single step LED on teach pendant shows the status for the original program execution.
- If the original program is paused after the RESUME_PROG instruction and then the operator moves the cursor to another line, the Resume Program is not executed at the next program execution.
- The status of the Resume Program execution is not displayed at the monitor screen in the program EDIT screen.
- The status line does not indicate when the Resume Program is executing.
- For a multi-tasking system, when the alarm code monitor feature is disabled and the approval DI is not defined, if the HOLD key is pressed, both parent and child task are paused.

10.20.4 I/O Interface

The Error Recovery sequence can be monitored and controlled remotely using digital I/O. The following I/O signals are available for use with Error Recovery.

- Approval DI
- Incomplete End DO
- Reset DI
- Status DO
- Maint DO

These signals can be assigned in the Error Recovery Setup screen described in Section 10.20.5. Refer Section 10.20.9 for more information on the I/O timing sequence.

Approval DI

If this input is defined (not zero), it is checked before executing error recovery.

- If Approval DI is ON, error recovery is approved, and Resume Programs or Maintenance Programs are executed at the appropriate times.
- If Approval DI is OFF, error recovery is not approved and Resume Programs or Maintenance Programs are not executed.

Incomplete End DO

When an Error Recovery program is aborted before its normal completion, the Incomplete End DO is turned ON. This DO is turned OFF at the next program execution. The Incomplete End DO is not set if the original program is aborted.

Check the status of this digital output signal before you input the START signal. If this signal is ON, confirm the current robot position. If an interference exists between the current robot position and the paused position of the original program, jog the robot to the position near the paused position before you input the START signal.

Reset DI

When the Incomplete End DO is used as a condition for a start input in the PLC, you need to turn off the Incomplete End DO remotely. When the Reset DI is input, the Incomplete End DO is turned OFF. After the operator performs the appropriate operation (for example, moves the robot to the position near the paused position of the original program), input this DI signal.

Status DO and Maint DO

The Status DO and Maint DO signals are provided to indicate whether an error recovery program will execute at the next START input signal.

- When the Status DO input signal is ON, it indicates that a Resume Program will execute at the next START input.
- When the Maint DO input signal is ON, it indicates that a Maintenance Program will execute at the next START input.
- When both the Status DO input signal is OFF and the Maint DO input signal is OFF, the original program will execute at the next START input.

The Status DO and Maint DO signals are very useful; without them, it is difficult to know which program will execute when resuming a paused program. For example, if you have set up Alarm Code Monitoring, only *some* errors will invoke Error Recovery, *not all*. As another example, if you have set up the Approval DI, its state dictates which program will execute.

NOTE Single step mode must be disabled prior to beginning error recovery. If single step mode is enabled during error recovery, it is ignored until the recovery sequence completes.

10.20.5 Setup

You set up Error Recovery for either Resume Program or Maintenance Program execution using the Error Recovery Setup screen shown in Figure 10-78. The items on this screen are listed and described in Table 10-23.

Figure 10-78. Error Recovery Setup Screen

Error Recovery Set		JOINT	100%
			1/12
Error recovery function common setup			
1	Error recovery function:	DISABLED	
2	Approval DI index No.:	0	
3	Incomplete end DO index No.:	0	
4	Reset DI index No.:	0	
5	Automatic start feature:	DISABLED	
RESUME PROGRAM type recovery			
6	Status DO index No.:	0	
7	Auto start Max count:	2	
8	Auto start Max count R[]:	0	
MAINTENANCE PROGRAM type recovery			
9	Fast exit/entry feature:	DISABLED	
10	Dry run exit/entry:	DISABLED	
11	Maintenance program:	DISABLED	
12	MAINT DO index No.:	0	
[TYPE] ALARM DI_ALARM ENABLED DISABLED			

Table 10-23. Error Recovery Setup Items

ITEM	DESCRIPTION
Error Recovery Function Common Setup	
Error Recovery Function	This item enables and disables Error Recovery.
Approval DI Index No.*	This item defines a digital input for approval of error recovery program execution.
Incomplete End DO Index No.*	This item defines a digital output to indicate that an error recovery program has been aborted before completion.
Reset DI Index No.*	This item defines a digital output for resetting the "Incomplete end DO."
Automatic Start Feature	This item enables and disables the automatic start feature.
RESUME PROGRAM Type Recovery	
Status DO Index No.	This item defines a digital output to indicate if a Resume Program or the original program will execute with the next start input.
Auto Start Max Count	This item defines the number of times Error Recovery is attempted for a given fault at the same location.
Auto Start Max Count R[]	This item defines the register number used for counting the number of times the error recovery program is started automatically.
MAINTENANCE PROGRAM Type Recovery	
Fast Exit/Entry Feature	This item enables and disables Maintenance Program error recovery.
Dry Run Exit/Entry	This item enables and disables the use of dry run speeds during exit and entry operations.
Maintenance Program	This item defines the name of a default Maintenance Program to be run when a teach pendant program has not executed a MAINT_PROG instruction.
MAINT DO Index No.	This item defines a digital output to indicate if a Maintenance Program or the original program will execute with the next start input.

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

Use Procedure 10-23 to set up Error Recovery items.

Procedure 10-23 Setting Up Error Recovery Items

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select Err recovery. You will see a screen similar to the following.

NOTE Items 9-12 are displayed only if the system variable \$RSMFST_SV.\$ffast_dsp = TRUE. If this variable is FALSE, these items are not displayed and Maintenance Program recovery is disabled.

Error Recovery Set	JOINT	100%
		1/12
Error recovery function common setup		
1 Error recovery function:	DISABLED	
2 Approval DI index No.:	0	
3 Incomplete end DO index No.:	0	
4 Reset DI index No.:	0	
5 Automatic start feature:	DISABLED	
RESUME PROGRAM type recovery		
6 Status DO index No.:	0	
7 Auto start Max count:	2	
8 Auto start Max count R[]:	0	
MAINTENANCE PROGRAM type recovery		
9 Fast exit/entry feature:	DISABLED	
10 Dry run exit/entry:	DISABLED	
11 Maintenance program:	DISABLED	
12 MAINT DO index No.:	0	
[TYPE]	ALARM	DI_ALARM ENABLED DISABLED

- 5 Move the cursor to each item and set it as desired. Refer to Table 10-23.

NOTE The default configuration of Error Recovery assumes that control is from the UOP. Error recovery can be configured to execute from the standard operator panel by setting the system variable \$RSMDRG_SV.\$chk_remote = FALSE.

CAUTION

If you configure error recovery to execute from the operator panel, there is no way to know that RESUME_PROG will be executed at the next start input unless you view the Error Recovery Status screen in the MANUAL FUNCTIONS screen.

Alarm Code Monitoring

You can set up Error Recovery to execute after either **all** errors or after a specific set of user-specified errors. Use the ALARM function key (Procedure 10-24) to define a list of specific errors. If you do not define any specific errors, then all PAUSE severity errors will start Error Recovery program execution.

When you have specified alarms to be monitored, and a fault occurs that is not on the alarm code monitor list, the original program will pause and the Resume Program will not be executed.

The default maximum number of alarms that can be monitored is 10. You can change this number by setting the value of \$RSMPRG_SV.\$NUM_ALARM and turning off the controller and then turning it on. The maximum value is 32.

NOTE If specific “Monitored alarm code” faults are not defined (they are all zero) and error recovery is enabled (Approval DI is ON), then all PAUSE severity errors will cause resume program execution at a START signal.

Use Procedure 10-24 to set up alarm code monitoring.

Procedure 10-24 Setting Up Alarm Code Monitoring

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select Err recovery. You will see a screen similar to the following.

Error Recovery Set		JOINT	100%
			1/12
Error recovery function common setup			
1	Error recovery function:	DISABLED	
2	Approval DI index No.:	0	
3	Incomplete end DO index No.:	0	
4	Reset DI index No.:	0	
5	Automatic start feature:	DISABLED	
RESUME PROGRAM type recovery			
6	Status DO index No.:	0	
7	Auto start Max count:	2	
8	Auto start Max count R[]:	0	
MAINTENANCE PROGRAM type recovery			
9	Fast exit/entry feature:	DISABLED	
10	Dry run exit/entry:	DISABLED	
11	Maintenance program:	DISABLED	
12	MAINT DO index No.:	0	
[TYPE] ALARM DI_ALARM ENABLED DISABLED			

Digital Input Alarms

The Error Recovery option allows you to define digital input signals that will generate user alarms. These user alarms can be monitored as error code 12278, as illustrated in Procedure 10-24 . Use Procedure 10-25 to set up digital input alarms.

You set user alarm information on the Setting User Alarm screen, shown in Figure 10-79. Refer to Section 3.17 for more information on User Alarm setup.

Figure 10-79. Setting User Alarm Screen

Setting/User Alarm		JOINT 10%
Alarm No.	User Message	1/10
[1]:	[Remote Error via DI[1]]	
[2]:	[]	
[3]:	[]	
[4]:	[]	
[5]:	[]	
[6]:	[]	
[7]:	[]	
[8]:	[]	
[9]:	[]	
[TYPE]		

Procedure 10-25 Setting Up Digital Input Alarms

- Step 1** Press MENUS.
- 2** Select SETUP.
- 3** Press F1, [TYPE].
- 4** Select Err recovery. You will see a screen similar to the following.

Error Recovery Set		JOINT	100%
			1/12
Error recovery function common setup			
1	Error recovery function:	DISABLED	
2	Approval DI index No.:	0	
3	Incomplete end DO index No.:	0	
4	Reset DI index No.:	0	
5	Automatic start feature:	DISABLED	
RESUME PROGRAM type recovery			
6	Status DO index No.:	0	
7	Auto start Max count:	2	
8	Auto start Max count R[]:	0	
MAINTENANCE PROGRAM type recovery			
9	Fast exit/entry feature:	DISABLED	
10	Dry run exit/entry:	DISABLED	
11	Maintenance program:	DISABLED	
12	MAINT DO index No.:	0	
[TYPE] ALARM DI_ALARM ENABLED DISABLED			

- 5** Press F3, DI_ALARM. See the following screen for an example.

Error Recovery Set SW		JOINT	100%
			1/3
	UALM	Severity	Type
			Value
1	[1]	LOCAL	DI[1] ON
2	[0]	LOCAL	DI[0] ON
3	[0]	LOCAL	DI[0] ON
[TYPE] DONE HELP			

- 6** Select and set the items as desired.
- 7** To display help information, press F5, HELP.
- 8** When you are finished setting DI_ALARM information, press F4, DONE, to return to the previous screen.

10.20.6 Programming

You use the following teach pendant instructions to specify appropriate recovery program names when you use Error Recovery:

- Resume Program instructions
- Maintenance Program instructions

Resume Program Instructions

The auto error recovery function executes the resume program defined in the teach pendant program. To define which resume program is used, use the RESUME_PROG instruction. To clear the resume program, use the CLEAR_RESUME_PROG instruction. See Figure 10-80 and Figure 10-81.

RESUME_PROG

Figure 10-80. RESUME_PROGRAM Instruction

```
RESUME_PROG = resume program name
```

CLEAR_RESUME_PROG

Figure 10-81. CLEAR_RESUME_PROG Instruction

```
CLEAR_RESUME_PROG
```

Figure 10-82 contains an example production program that sets RESUME_PROG to WIRE_CUT, which is shown in Figure 10-83.

Figure 10-82. WELD.TP Example Program

```
1: J P[1] 40% FINE
2: RESUME_PROG = WIRE_CUT
3: L P[2] 300mm/sec FINE ARC START[1]
4: L P[3] 50cm/min CNT100
5: L P[4] 50cm/min FINE ARC END[2]
6: CLEAR_RESUME_PROG
7: L P[5] 300mm/sec FINE
```

Figure 10-83. WIRE_CUT.TP (Resume Program) Example Program

```
1: L P[10] 50mm/sec FINE INC
2: PR[1]=LPOS
3: J P[11] 50% FINE
4: WO[4] = ON pulse 0.5sec Feed wire
5: L P[12] 20mm/sec FINE
6: WAIT 0.8sec Wait for completion
7: L P[11] 20mm/sec FINE of cutting wire
8: RESUME_PROG = WIRE_CUT2
9: J PR[1] 50% FINE
```

Figure 10-82 shows how to define the resume program. The WIRE_CUT program is defined as the resume program in line 2 using the RESUME_PROG instruction. The WIRE_CUT program is cleared from the resume program in line 6 of WELD.TP using the CLEAR_RESUME_PROG instruction. Therefore, the WIRE_CUT program is available as the resume program only during program lines 3, 4, and 5 in WELD.TP.

When the automatic start feature is enabled and WELD.TP is paused by a monitored alarm and resumed on lines 3, 4, or 5, the WIRE_CUT program is executed as the resume program and the wire is automatically cut using the WIRE_CUT program. In the program WELD.TP, the resume program is not executed after line 6.

NOTE In the WIRE_CUT.TP program, PR[1] is near the fault position. The INC (incremental) motion option in line 1 puts PR[1] near the fault position by the value of the INC position. In general, the INC position will be a z offset, such as P[10]: 0, 0, 25, 0, 0, 0.

NOTE In WIRE_CUT.TP, line 8, the RESUME_PROGRAM is redefined to be WIRE_CUT2.TP. If another error occurs in WELD.TP after executing the recovery sequence, WIRE_CUT2.TP will be executed upon program resume instead of WIRE_CUT.TP.

**WARNING**

If the wrong program is defined as the resume program, the robot will move toward an unexpected place. Be sure to define the correct resume program. Otherwise, you could injure personnel or damage equipment.

Maintenance Program Instructions

To define which maintenance program is used, use the MAINT_PROG instruction. To disable the ability to use the return path, use the RETURN_PATH_DSBL instruction. See Figure 10-84 and Figure 10-85.

MAINT_PROG

Figure 10-84. MAINT_PROGRAM Instruction

MAINT_PROG = maintenance program name

RETURN_PATH_DSBL

Figure 10-85. RETURN_PATH_DSBL Instruction

RETURN_PATH_DSBL

Figure 10-86 contains an example production program that sets RESUME_PROG to WIRE_CUT, which is shown in Figure 10-83.

Figure 10-86. WELD.TP Example Program

```

1: J P[1] 40% FINE
2: RESUME_PROG = WIRE_CUT
3: L P[2] 300mm/sec FINE ARC START[1]
4: L P[3] 50cm/min CNT100
5: L P[4] 50cm/min FINE ARC END[2]
6: CLEAR_RESUME_PROG
7: L P[5] 300mm/sec FINE

```

Programming Procedure Use Procedure 10-26 to add Error Recovery instructions to a program.

Procedure 10-26 Adding Error Recovery Instructions to a Program

- Condition** ■ You are editing a teach pendant program.
- Step**
- 1 Press F1, [INST], to display the list of instructions.
 - 2 Select 8 — next page —, to display more instructions.
 - 3 Select Program control. You will see a screen similar to the following.

Instruction	JOINT	100%
1 PAUSE	5 CLEAR_RESUME_PROG	
2 ABORT	6 RETURN_PATH_DSBL	
3 ERROR_PROG	7 MAINT_PROG	
4 RESUME_PROG	8	
PROGRAM		

- 4 Select RESUME_PROG, CLEAR_RESUME_PROG, RETURN_PATH_DSBL, or MAINT_PROG.

10.20.7 Testing

Normally you use Error Recovery when the teach pendant is disabled during production operation. However, when you define a recovery program or test a production program, you might want to execute the recovery program even though all conditions for execution have not been met.

You can test error recovery with the teach pendant enabled from the Auto Error Recovery Manual Function screen when you select **TP_TEST** as the operation mode. Refer to the **Operation mode** item in Section 10.20.8.

10.20.8 Error Recovery Manual Function

You can use the Manual Function screen to do the following:

- Display the status of error recovery status DO
- Display the resume program name defined by the original program
- Select the operation mode
- Monitor the conditions related to the status of the error recovery status DO

Table 10-24 lists and describes the items on the Manual Function screen.

Table 10-24. Auto Error Recovery Manual Function Screen Items

ITEMS	DESCRIPTION
Error recovery DO status	The status of the error recovery status DO is displayed in this field regardless of whether the error recovery status DO is defined in the Auto Error Recovery Setup screen. NOTE: Status DO refers to an internal signal, not a digital output signal. When a digital output is configured, this internal signal will be reflected in the digital output.
Defined resume program	The resume program name defined by the original program is displayed.
Operation mode Default: AUTO	The operation mode has the following three modes. The default mode is AUTO and it is automatically changed to AUTO when you exit from this screen. <ul style="list-style-type: none"> ● AUTO This mode should be selected when the teach pendant is disabled. When this mode is selected, the Resume Program is executed according to the status of the alarm code feature and the error recovery approval DI feature. When this mode is selected and the teach pendant is enabled, the resume program is not executed but the original program is executed when SHIFT FWD is pressed. ● NOEXEC When this mode is selected, the error recovery status DO is always turned off regardless of whether the teach pendant is enabled. This means that the Resume Program is not executed by the next program execution. ● TP_TEST This mode should be selected when the teach pendant is enabled. When this mode is selected and the teach pendant is enabled, even if the alarm code feature and the error recovery approval DI feature are not satisfied, the error recovery status DO is turned on. This means the Resume Program is always executed by the next program execution.

Manual Function Detail Screen

In the Manual Function DETAIL screen you can monitor recovery DO status, recovery program name, and operation mode related to the status of the error recovery status DO.

When F2, DETAIL, is pressed on the Auto Error Recovery Manual Function screen, the conditions related to the error recovery status DO are displayed.

When all elements are Yes or None (not used), the error recovery status DO is turned ON.

When the error recovery status DO is OFF and you are not sure of the cause, check the information on this screen.

Refer to Table 10-25 for a description of the items on the Manual Function DETAIL screen.

Table 10-25. Auto Error Recovery Manual Function Detail Screen Items

ITEMS	DESCRIPTION
Auto error recovery enabled	This item shows whether the Error Recovery function is enabled in the auto error recovery setup screen.
PAUSED & resume prog incomp	This item shows the following: <ul style="list-style-type: none"> • The original program exists. • The original program is paused. • The execution of the resume program selected by the original program has not been completed.
Program has motion group	This item shows whether the original program has motion control.
Not in single step mode	This item shows whether the single step mode is disabled. The single step LED on the teach pendant specifies the status of single step for the original program (\$TP_DEFPROG). When the Resume Program is paused and then single step LED is turned on, the error recovery DO remains ON because single step for the Resume Program is disabled.
Resume program is defined	This item shows whether the resume program is defined by the original program.
Mode is (xxxxxx)	This item shows whether the operation mode is the desired one for this current situation. If the teach pendant is disabled, AUTO is displayed in the field "xxxxxx". If the teach pendant is enabled, TP_TEST is displayed in the field.
Approval DI is ON	This item shows whether the status of the error recovery approval DI is ON. If the index of this DI is not defined or the teach pendant is enabled, "None" is displayed.
Defined alarm occurs	This item shows that the defined alarm has occurred and that the original program has been paused by the alarm, if the alarms are defined in the setup screen. If the alarm code is not defined or the teach pendant is enabled, "None" is displayed.
Remote when \$RMT_MASTER is 0	This item shows whether the remote condition is satisfied. This feature is available only when the teach pendant is disabled and \$RMT_MASTER is 0 and \$RSMPRG_SV.\$CHK_REMOTE is TRUE. If you want to remove the remote condition, you can set \$RSMPRG_SV.\$CHK_REMOTE to FALSE.
No disabled options	The Error Recovery function has the restriction of non-coexistence. It shows whether the non-coexistent options exist in the software. Refer to Section 10.20.3.
User condition param enable	It shows whether the user condition parameter (system variable) is TRUE. The default is TRUE. This system variable is \$AUTORCV_ENB.

Use Procedure 10-27 to perform manual operation of Error Recovery.

Procedure 10-27 Manual Operation of Error Recovery

- Step**
- 1 Press MENUS.
 - 2 Select MANUAL FCTNS.
 - 3 Press F1, [TYPE].
 - 4 Select Err recovery. You will see a screen similar to the following.

Error Recovery MNFC		JOINT	100%
			1/1
Error recovery DO status:			OFF
Defined resume program:			WIRE_CUT
1 Operation mode:			AUTO
[TYPE]	DETAIL		[CHOICE]

- 5 Select the mode you want to use. You can change operation mode between AUTO, NOEXEC, and TP_TEST by pressing F4, [CHOICE]. During production, this mode should be AUTO.
- 6 Press F2, DETAIL, and the following information screen for the error recovery status DO is displayed.

Error Recovery MNFC		JOINT	100%
			1/11
1 Auto error recovery enabled:			Yes
2 PAUSED & resume prog incomp:			Yes
3 Program has motion group:			Yes
4 Not in single step mode:			No
5 Resume program is defined:			Yes
6 Mode is (AUTO):			Yes
7 Approval DI is ON:			None
8 Defined alarm occurs:			Yes
9 Remote when \$RMT_MASTER is 0:			Yes
10 No disabled options:			No
11 User condition param enable:			Yes
[TYPE]			DONE

NOTE Items on this screen cannot be changed on this screen.

10.20.9 I/O Timing Sequence

See Figure 10-87 through Figure 10-91 for timing diagrams.

Figure 10-87. Normal Operation Auto Start Mode

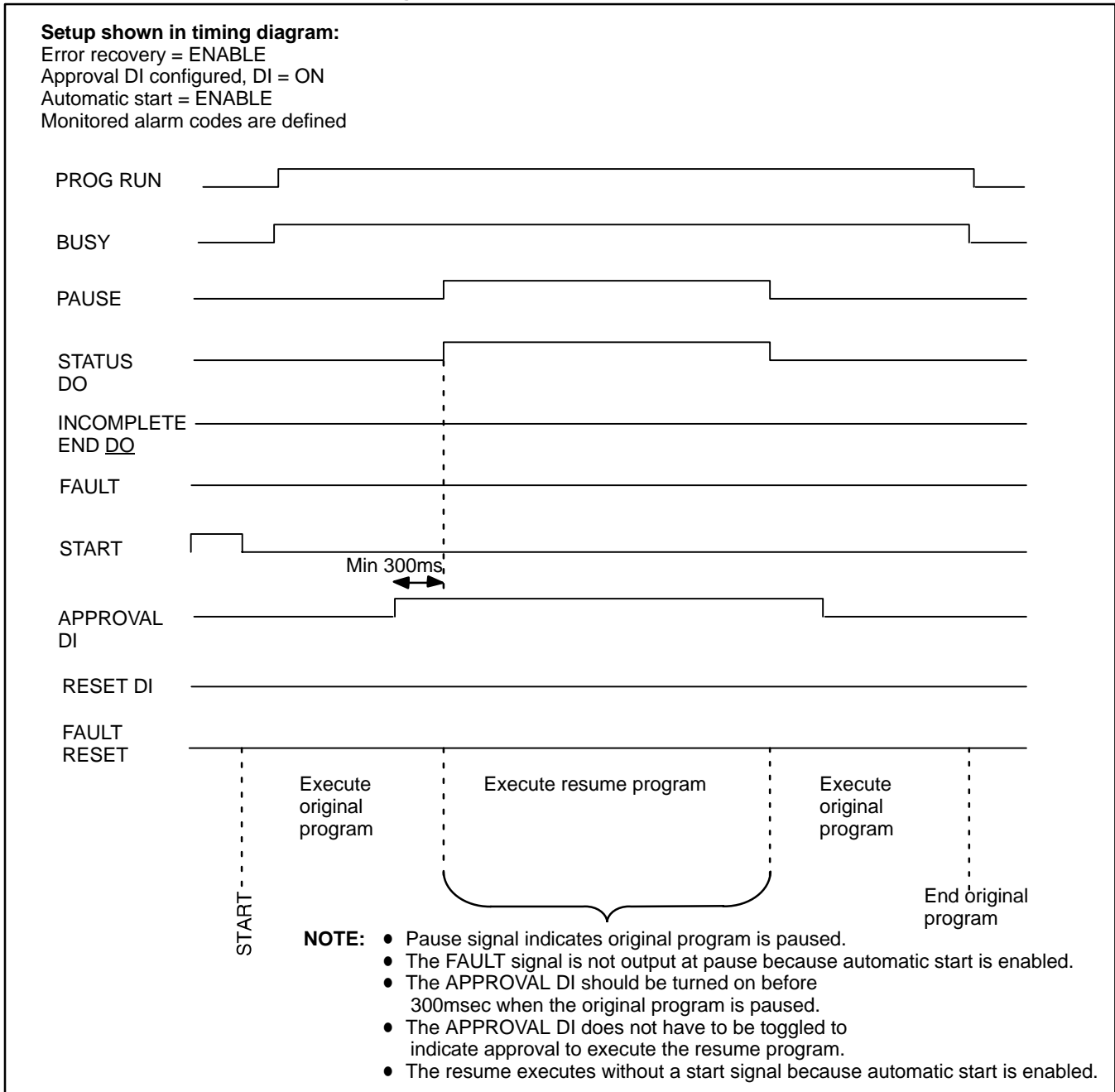


Figure 10-88. Normal Operation without Execution of Resume Program

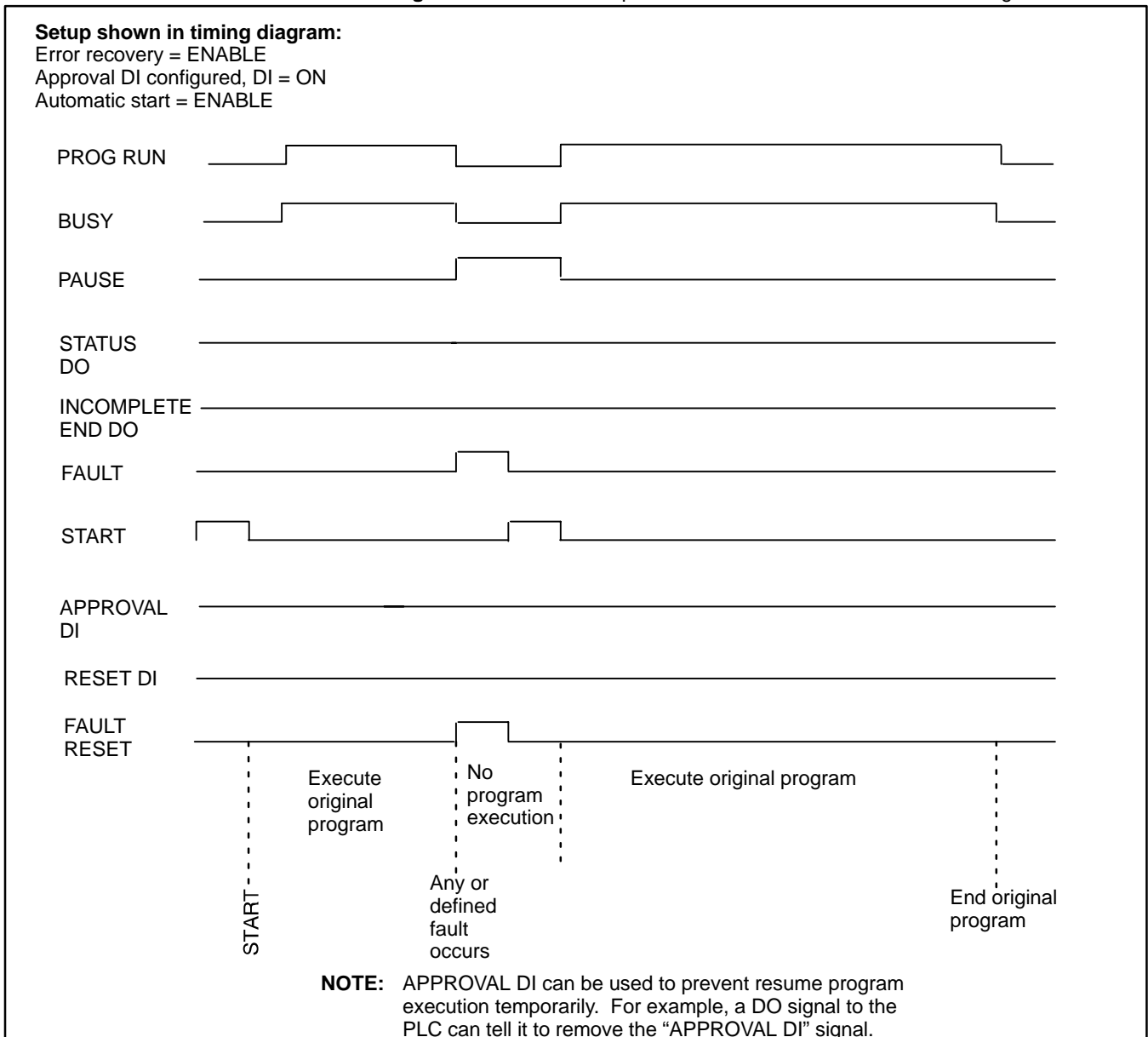


Figure 10-89. Resume Program Aborted

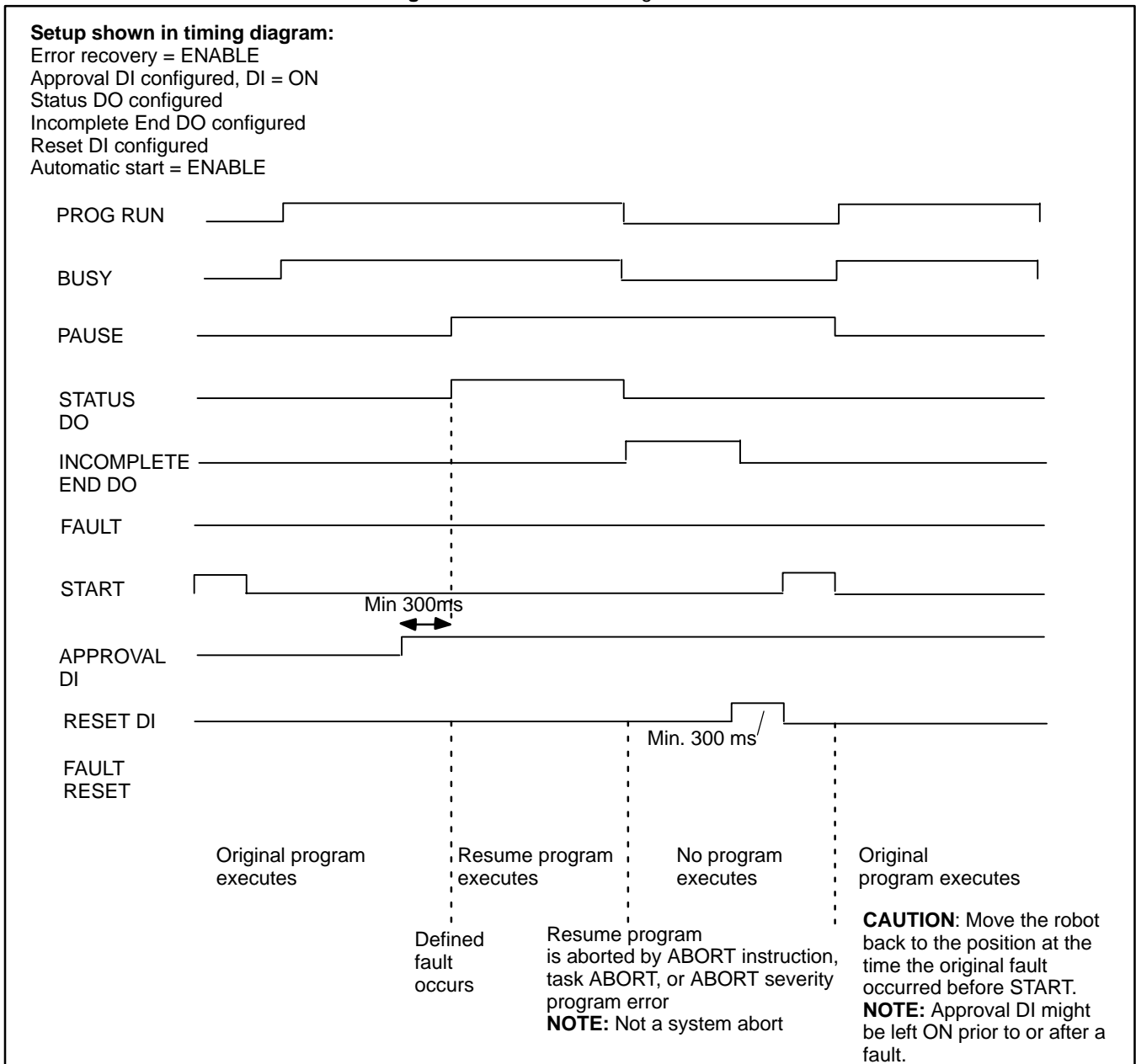


Figure 10-90. Normal Operation (Automatic Start DISABLED)

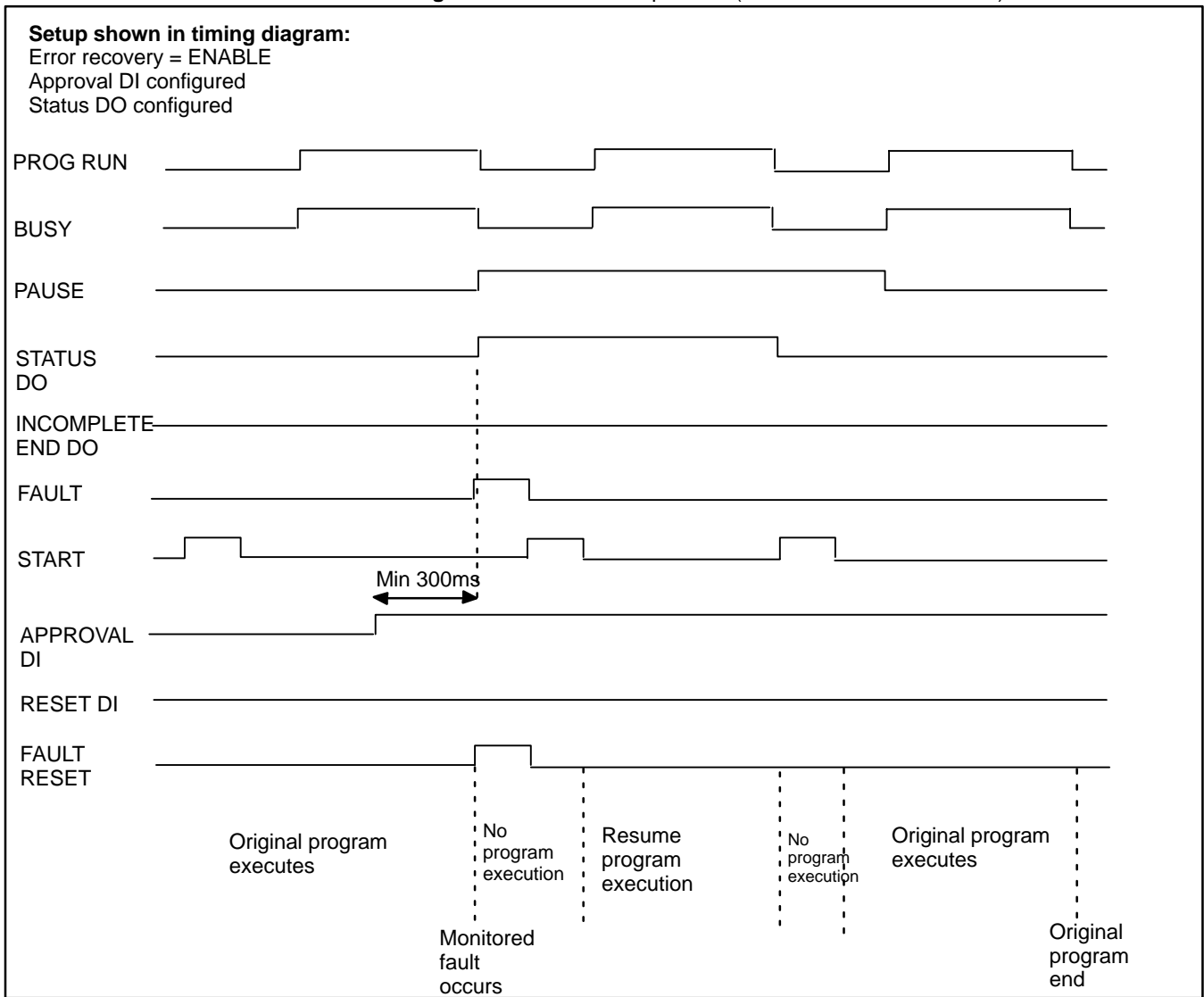
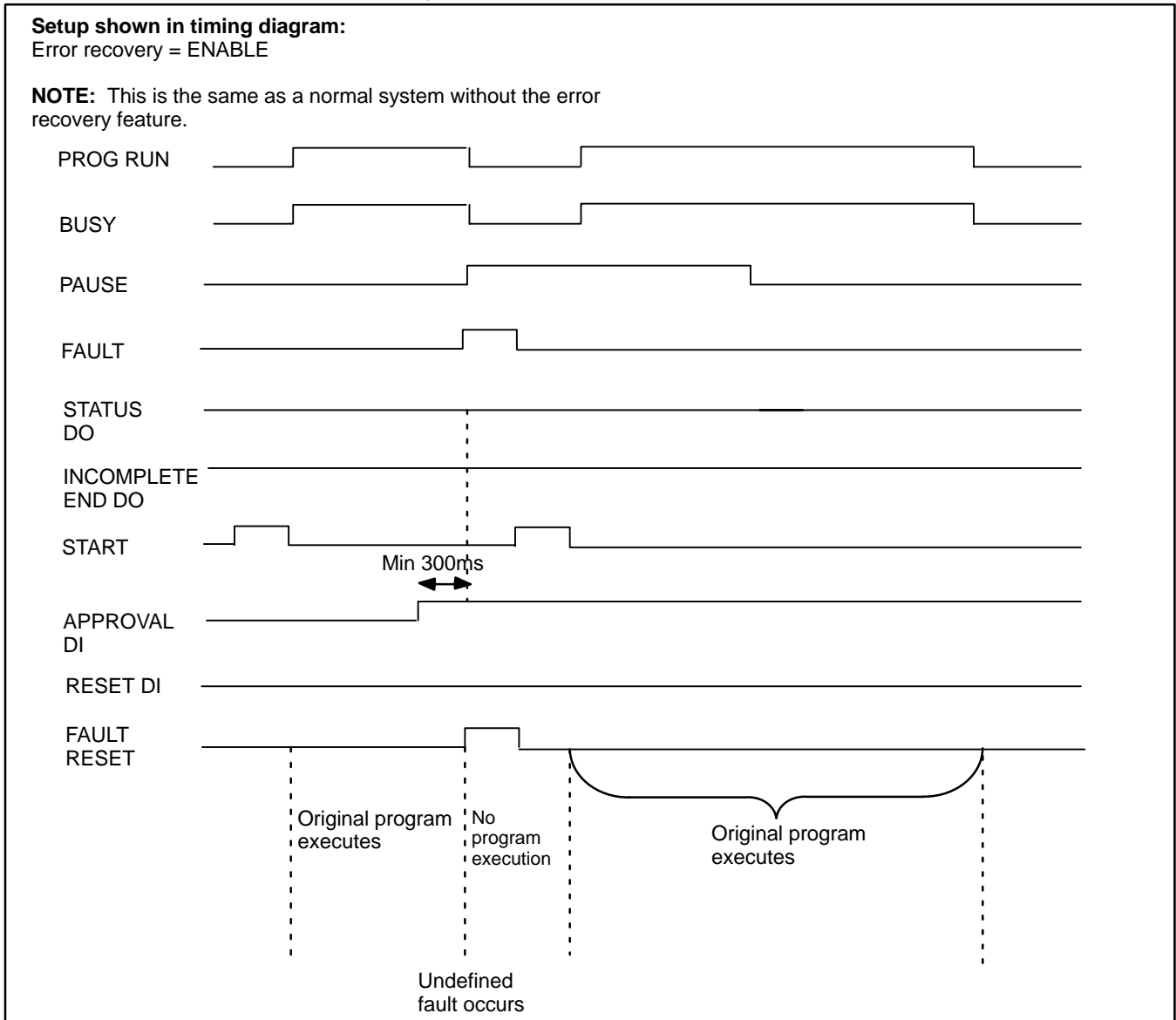


Figure 10-91. Auto Mode When an Undefined Alarm Occurs



10.21 AUTO NORMAL UTILITY (OPTION)

The Auto Normal utility is a software option that is used to align the end-of-arm tooling of the robot perpendicular, or normal, to a surface, such as a workpiece. This option assumes that a height or range sensor is available on the end of the robot arm.

FANUC Robotics has successfully used the Auto Normal utility in laser cutting applications in which a laser height sensor is integral with the cutting tool on the end of the robot arm. This cutting tool with the sensor can be purchased from FANUC Robotics. You can also use this option with your own range or height sensors. Be sure to consider carefully the mounting and usage of the height sensor so you can use the Auto Normal utility effectively. Contact FANUC Robotics for further assistance.

The Auto Normal utility provides a setup screen in which you can enter various search parameters to tune the utility to your particular application. The search parameters are used with a proprietary algorithm to move the robot in a particular manner relative to the surface, such as the workpiece. At the end of these motions, the robot will re-orient the end of the robot arm to be normal to the surface. The jog frame of the robot is also set up automatically so that when you jog the robot in the X and Y directions, the end of the robot arm maintains its perpendicular orientation relative to the surface of the workpiece.

Without the Auto Normal utility, it would be time consuming to define an accurate relative position and orientation. However, in applications such as laser cutting of shapes, the tool location and orientation are critical for cut quality.

The Auto Normal utility does the following:

- Simplifies the teaching of robot positions
- Reduces teaching time
- Improves the accuracy of taught positions relative to the work surface

To use the Auto Normal utility option, you do the following:

1. Set up Auto Normal – Section 10.21.2
2. Execute Auto Normal to perform the search and yield the appropriate positions and frames for accurate teaching – Section 10.21.3

This section contains information on these operations and also includes background information you need to know to use the Auto Normal utility.

10.21.1 Overview

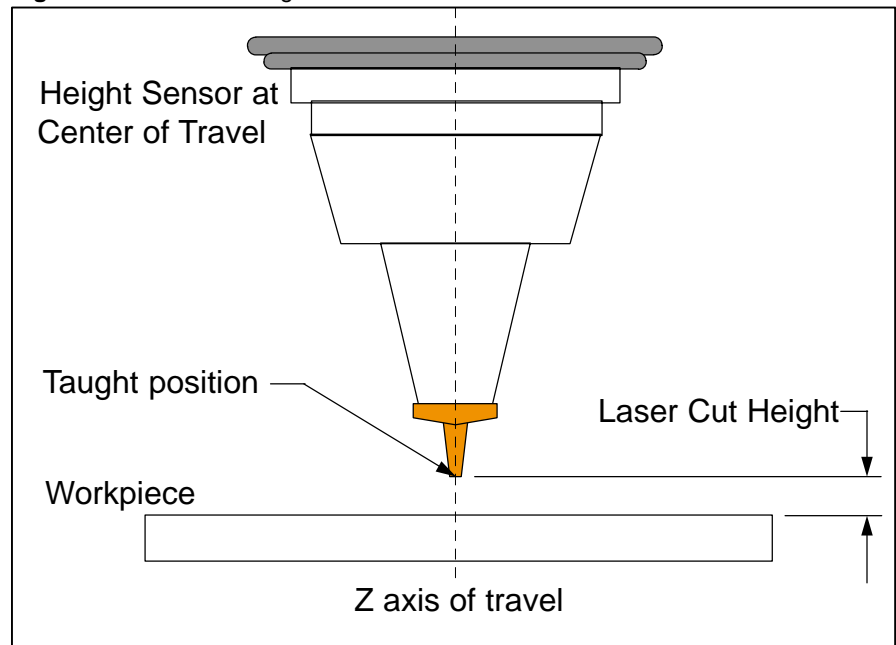
This section contains information on the following topics, which you must know in order to use the Auto Normal utility effectively. The descriptions in this section use a laser height sensor as an example. You can use your own height sensor in a similar manner.

- Laser height sensor
- Search motion
- Shape frames

Laser Height Sensor

The laser cutting tool has an integrated height sensor. The laser height sensor uses an inductive proximity sensor to locate, within a certain degree of accuracy, a surface within the sensor range. During the searching utility, the height sensor will be locked to its center of travel. The final position will be at a known distance (called the laser cut height) above the workpiece controlled by the laser controller. See Figure 10-92.

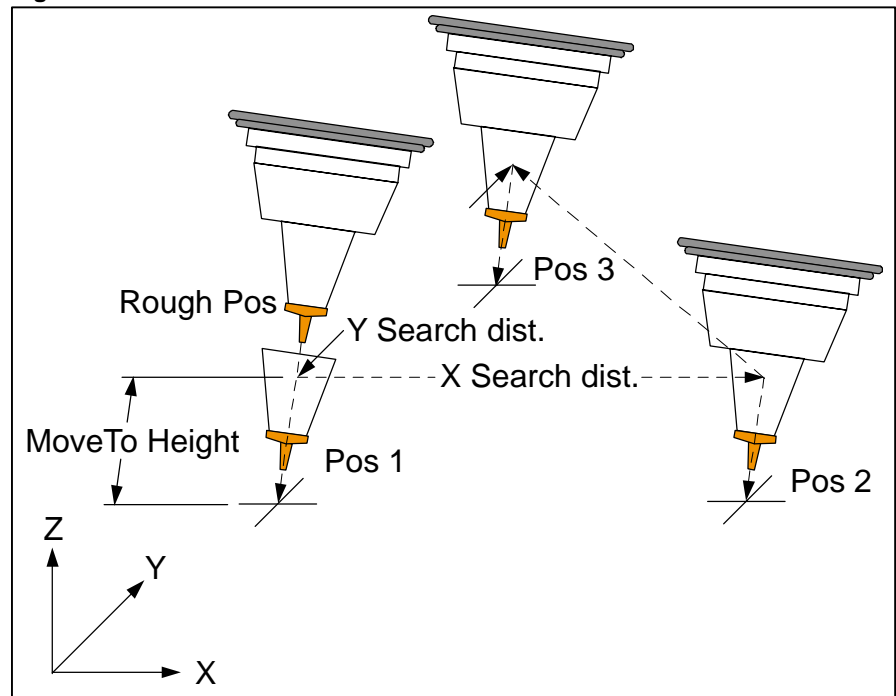
Figure 10-92. Laser Height Sensor



Search Motion

The Auto Normal utility moves the robot, in a specified manner, from a rough position toward an object in the TOOL Z direction. The robot stops searching and moving when the sensor detects an object. A three-point search algorithm is used to locate the object and to determine its orientation and distance relative to the robot. See Figure 10-93.

Figure 10-93. Search Motion

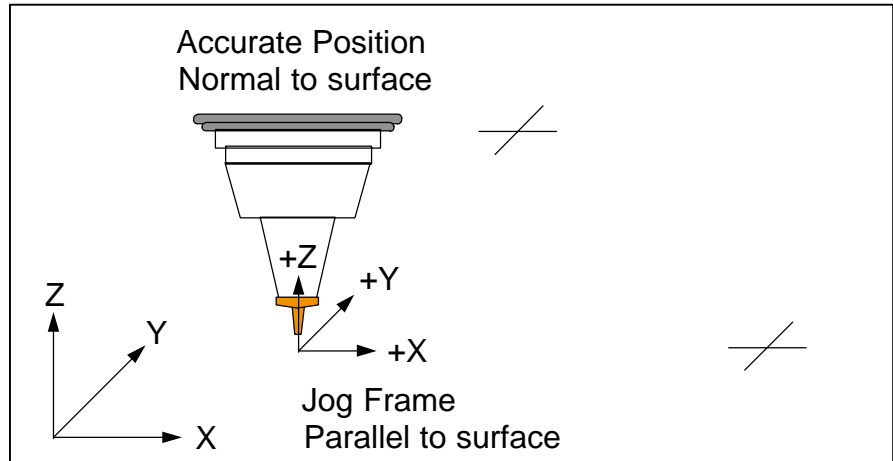


Three **search sizes** are available: small, medium, and large. A larger search on a flat surface will produce a more accurate normalized position. However, smaller search sizes will allow you to perform a search within a small area where tight teaching tolerances are required.

After a search has been completed, the robot will be located at a new position that is normal (or perpendicular) to the surface where a teach pendant position can be recorded or touched up. The cutting torch will be located at a known distance above the workpiece, determined by the laser controller.

A **jog frame** is now set, where the X-Y plane is parallel to the surface of the *shape frame*. This reduces the amount of teaching time by allowing you to jog the robot parallel to the surface and record additional positions located along the same surface. See Figure 10-94. Refer to the next section (“Shape Frames”) for more information on shape frames.

Figure 10-94. Positions and Frames Defined After the Search



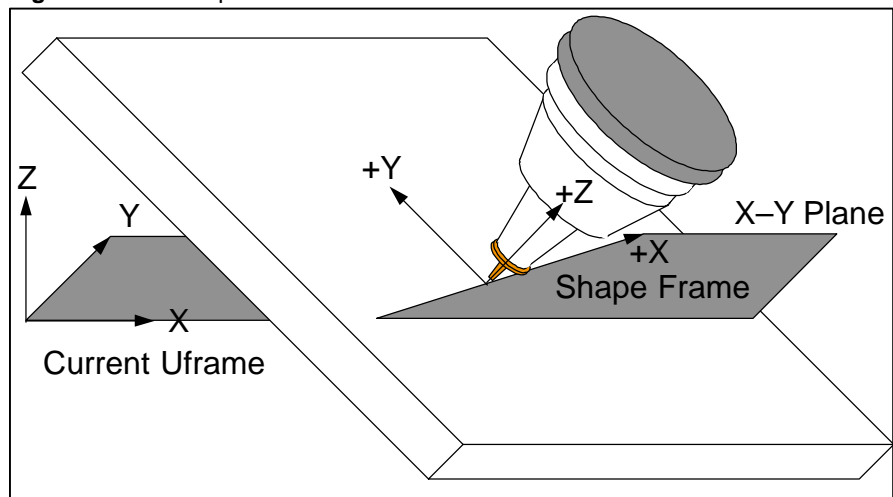
Shape Frames

A shape frame is calculated from a shape center position and current uframe:

- The shape frame **X-direction** is parallel to the object surface where the current UFRAME X-Y plane intersects the object at the shape center position.
- The shape frame **Y-direction** is also parallel to the object surface but perpendicular to the X-direction.
- The shape frame **Z-direction** is into the tool along the TOOL Z vector.

See Figure 10-95.

Figure 10-95. Shape Frames



10.21.2 Setup

You set up the Auto Normal utility by setting items on the SETUP Auto Normal screen. Table 10-26 lists and describes the items you must set. Use Procedure 10-28 to set up Auto Normal.

Table 10-26. Auto Normal Setup Items

ITEM	DESCRIPTION
Sensor Input	This item allows you to specify the number of the digital input that the robot will receive when the sensor locates an object.
Search Speed default: 20 mm/sec min: 1 mm/sec max: 20 mm/sec	This item allows you to specify the speed the robot will use to search toward the object. In general, slower search speeds result in more accurate results.
MoveTo Speed default: 200 mm/sec min: 1 mm/sec max: 200 mm/sec	This item allows you to specify the travel speed the robot will use while moving between search positions.
Search Distance default: 20 mm min: 1 mm max: 50 mm	This item allows you to specify the maximum distance the robot will search toward an object before the robot ends the search and posts an error message.
MoveTo Height default: 10 mm min: 1 mm max: 50 mm	This item allows you to specify the height above an object at which the robot will travel to and from search positions.
Current Search Size default: SMALL values: SMALL, MEDIUM, LARGE	This item allows you to specify the currently selected search size in the X-Y plane: <ul style="list-style-type: none"> ● SMALL – X = 10 mm; Y = 10 mm ● MEDIUM – X = 15 mm; Y = 15 mm ● LARGE – X = 20 mm; Y = 20 mm In general, larger search areas result in more accurate normal positions
Small X default: 10 mm Small Y default: 10 mm Medium X default: 15 mm Medium Y default: 15 mm Large X default: 20 mm Large Y default: 20 mm min: 10 mm max: 200 mm	These items allow you to define the X and Y distances between search positions for each kind of search size: small, medium, and large.

Procedure 10-28 Setting Up Auto Normal

- Step**
- 1 Press MENUS.
 - 2 Select SETUP.
 - 3 Press F1, [TYPE].
 - 4 Select Auto Normal. You will see a screen similar to the following.

SETUP Auto Normal		JOINT 10 %
		1/9
1 Sensor input:		DI[13]
2 Search speed:		20 mm/sec
3 MoveTo speed:		200 mm/sec
4 Search distance:		10 mm/sec
5 MoveTo height:		20 mm
6 Current search size:		SMALL
[SEARCH]	X	Y
7 Small:	10.000	10.000 mm
8 Medium:	15.000	15.000 mm
9 Large:	20.000	20.000 mm
[TYPE]		

- 5 Move the cursor to each item and set as desired.

10.21.3 Execution

The Auto Normal utility uses a teach pendant macro to call a KAREL routine to execute the search. Use Procedure 10-29 to execute the Auto Normal utility.

The teach pendant macro is named AUTO NORMAL. the AUTO NORMAL macro executes the teach pendant program named NORMAL.TP. The NORMAL.TP teach pendant program contains one instruction, "CALL FINORM." FINORM is a KAREL program that contains all of the search motions.

Procedure 10-29 Executing Auto Normal to Perform a Search

- Step**
- 1 Hold the teach pendant and continuously press the DEADMAN switch on the back of the teach pendant.
 - 2 Turn the teach pendant ON/OFF switch to the ON position.

**WARNING**

In the next step, the robot will move. To stop the robot immediately any time during jogging, release the DEADMAN switch or press the EMERGENCY STOP button.

- 3 Press and hold the SHIFT key and press the MAN FCTNS key. After the robot starts moving, you can release the MAN FCTNS key but must **continue holding down** the SHIFT key.

NOTE If you release the SHIFT key while the robot is executing the search, the Auto Normal search sequence will be aborted. If this happens, jog the robot or step back to the desired center position and repeat Steps 1 through 3.

When the Auto Normal utility has completed the search sequence, a jog frame is set to jog the robot with x and y directions parallel to the workpiece along the shape frame.

10.22 TCP SPEED PREDICTION (OPTION)

The TCP Speed Prediction option provides the predicted value of the TCP speed continuously as the robot is moving. Some robot applications require knowing the Tool Center Point (TCP) speed of the robot ahead of time. This is especially useful for applications that dispense or spray material from a gun attached to the robot arm, where dispensing or spraying equipment cannot be turned on instantaneously. A delay, called the *equipment delay*, occurs before the flow of fluid becomes effective or before its rate, called the *flow rate*, changes. If the TCP speed is known ahead of time by the equipment delay time, the flow rate can be adjusted accordingly to increase or decrease depending upon the TCP speed. This provides better quality of dispensed material on the workpiece.

You must enable the TCP Speed Prediction option and set the equipment delay using the system variables described in this section. The TCP Speed Prediction software then computes the predicted value and sets it in another system variable. Your application can access these predicted values through teach pendant or KAREL programs and use them to control equipment precisely.

Although there are a few system variables related to this option, the following are the most important:

- \$TCPPIR.\$enable_tcp – enables or disables the TCP Speed Prediction option
- \$TCPPIR.\$tcdelay – sets the prediction or equipment delay time
- \$TCPSPPEED.\$speed – this is where the output or the TCP predicted value is output continuously

You use the TCP Speed Prediction option by setting system variables that control how the option works. This section contains the following information:

- An **overview** of the TCP Speed Prediction option to help you decide when you can use the option – Section 10.22.1
- TCP Speed Prediction **system variables** that you set to the appropriate values to use TCP Speed Prediction – Section 10.22.2

10.22.1 Overview

This section contains an overview of the TCP Speed Prediction option in the following areas:

- Software setup
- Execution
- Limitations

Software Setup

You must set several TCP Speed Prediction system variables properly in order to use TCP Speed Prediction. In particular, you must set the \$enable_tcp and \$tcdelay fields in the \$TCPPIR system variable structure properly. Refer to Section 10.22.2 for details.

Execution

After you have enabled TCP Speed Prediction using the system variables, TCP Speed Prediction runs automatically as a background task and does not require any additional user input or action. However, to take advantage of TCP Speed Prediction, an application task should be written that monitors the \$TCPSPPEED system variable structure (typically the \$speed field) for the predicted speed values, and performs whatever processing is required to generate application process control signals.

If the Remote TCP option is used, TCP Speed Prediction shifts its reference frame automatically as an application program switches to the Remote TCP mode and back again. Therefore, no user action is required. However, you must understand that the speed reference – the \$TCPSPPEED.\$vspeed and \$TCPSPPEED.\$speed variables in particular – are with respect to the Remote TCP frame instead of the WORLD frame during Remote TCP motions.

Limitations

The TCP Speed Prediction option has the following limitations:

- Prediction is supported only for group 1 motion. Multiple motion groups can exist and be programmed, but are not supported for TCP Speed Prediction.
- Prediction for non-integrated extended axes is not supported.
- Prediction for JOINT motions is approximated as if the motions were Cartesian motions. Such estimation is generally good enough if the JOINT motion is not too large. If predicted speed accuracy is important, linear motion is recommended.
- Prediction becomes invalid during instances of speed and motor limit conditions and other motion warnings and errors. In these cases, you could correct the programmed path to remove the error conditions. Prediction also becomes temporarily invalid during instances of dynamic speed override changes.
- You can change the prediction time delay (\$TCPPIR.\$tcdelay) on-the-fly. However, this might cause a momentary discontinuity in the value of the predicted speed output.

- TCP Speed Prediction has no knowledge of WAIT or other system delay instructions that might interrupt robot motion. Therefore, TCP Speed Prediction will continue to predict the speed based upon subsequent motion commands in these cases. However, TCP Speed Prediction will resynchronize itself properly after the delay condition has been met and motion resumes.
- TCP Speed Prediction is not compatible with any of the general tracking options, such as seam tracking and line tracking.
- Transitions between normal TCP Speed Prediction and Remote TCP Speed Prediction include a change of reference frame from WORLD frame to USER (REMOTE TCP) frame. When this transition takes place, the TCP Speed Prediction task will “blend” this change of reference frame. This might cause the predicted speed signal to show an apparent speed decrease during the blending period.

10.22.2 System Variables

Three system variable structures are generated when loading the TCP Speed Prediction option. These structures control the state and configuration of this option, and provide the speed and other output values for user program access. These variables are described in this section:

- \$TCPPIR structure
- \$TCPPSPEED structure
- \$TCPP_CFG structure

\$TCPPIR STRUCTURE

Name: TCP Speed Prediction Instruction Record System Variable Structure

Description: This set of variables allows the user to enable the TCP Speed Prediction softpart and set the equipment delay (prediction) time. The individual fields within this structure are defined below.

User Interface Location: SYSTEM Variables screen

\$TCPPIR.\$enable_tcp

Minimum:	FALSE	Default:	FALSE	Maximum:	TRUE
KCL/Data:	RW	Program:	RW	GET/SET_VAR:	RW Data Type: BOOLEAN

Name: TCP Speed Prediction Enable Switch

Description: This allows you to enable or disable the TCP Speed Prediction softpart. A predicted speed output is generated when the softpart is enabled. This variable is initially set FALSE at the time of loading, and must be set TRUE by the user prior to using the speed prediction values.

Power Up: This value initializes to FALSE and maintains its previous value over subsequent power cycles.

User Interface Location: SYSTEM Variables screen

\$TCPPIR.\$tcdelay

Minimum:	-5000.0	Default:	200.0	Maximum:	5000.0
KCL/Data:	RW	Program:	RW	GET/SET_VAR:	RW Data Type: REAL

Name: TCP Speed Prediction (Equipment) Delay Time (milliseconds)

Description: This value allows the user to specify the equipment delay time to be used as the prediction time by the TCP Speed Prediction softpart. This variable is initially set to 200 at the time of loading, but should be set to an appropriate positive value by the user prior to using the speed prediction values.

Power Up: This value initializes to 200 (msec) and maintains its previous value over subsequent power cycles.

User Interface Location: SYSTEM Variables screen

NOTE: Modified values of this variable will take effect immediately if \$TCPP_CFG.\$otf_tim_enb is TRUE, but will not take effect until all robot motion has stopped if \$TCPP_CFG.\$otf_tim_enb is FALSE.

\$TCPPSPEED STRUCTURE

Name: TCP Speed Prediction Speed Output System Variable Structure

Description: These are the output system variables that involve the predicted speed of the robot Tool Center Point (TCP). These variables are updated by the TCPP task softpart at the interval specified by the value of \$TCPP_CFG.\$tcpp_time. The individual fields within this structure are defined below.

User Interface Location: SYSTEM Variables screen

See Also: \$TCPP_CFG.\$tcpp_time

\$TCPPSPEED.\$accel

Minimum:	0	Default:	0	Maximum:	1000.0
KCL/Data:	RO	Program:	RO	GET/SET_VAR:	RO Data Type: REAL

Name: TCP Speed Prediction Acceleration (millimeters/second)

Description: This is the predicted acceleration of the robot TCP which corresponds to the value of the predicted speed (\$TCPPSPEED.\$speed). It is a directionless value, but its sign is valid (positive for increasing TCP speed, negative for decreasing TCP speed).

Power Up: This value initializes to 0 at each power cycle and is updated automatically by the TCPP task softpart at every TCPP cycle.

User Interface Location: SYSTEM Variables screen

See Also: \$TCPPSPEED.\$speed

\$TCPPSPEED.\$motype

Minimum:	6	Default:	6	Maximum:	8
KCL/Data:	RO	Program:	RO	GET/SET_VAR:	RO Data Type: INTEGER

Name: TCP Speed Prediction Programmed Motion Type

Description: This is the programmed motion type of the last active TCPP motion segment at the time of the most recent TCP Speed prediction update. The values are as follows:

- JOINT = 6
- LINEAR = 7
- CIRCULAR = 8

Power Up: This value initializes to 0 at each power cycle and is updated automatically by the TCPP task softpart at every TCPP cycle.

User Interface Location: SYSTEM Variables screen

\$TCPPSPEED.\$prog_speed

Minimum:	0.0	Default:	0.0	Maximum:	2000.0
KCL/Data:	RO	Program:	RO	GET/SET_VAR:	RO Data Type: REAL

Name: TCP Speed Prediction Programmed Speed

Description: This is the programmed speed of the last active TCPP motion segment at the time of the most recent TCP Speed prediction update.

Power Up: This value initializes to 0 at each power cycle and is updated automatically by the TCPP task softpart at every TCPP cycle.

User Interface Location: SYSTEM Variables screen

\$TCPPSPEED.\$speed

Minimum: .0.0 Default: 0.0 Maximum: 2000.0
 KCL/Data: RO Program: RO GET/SET_VAR: RO Data Type: REAL

Name: TCP Speed Prediction Speed (Absolute Velocity Magnitude) (millimeters / second)

Description: This is the predicted speed of the robot TCP computed as the absolute value of the magnitude of the predicted Cartesian velocity (\$TCPPSPEED.\$vspeed). It is referenced (without sign or direction) to the WORLD frame for normal prediction, and to the currently selected Remote TCP frame (also without sign or direction) for Remote TCP operation.

Power Up: This value initializes to 0 at each power cycle and is updated automatically by the TCPP task softpart at every TCPP cycle.

User Interface Location: SYSTEM Variables screen

See Also: \$TCPPSPEED.\$vspeed

\$TCPPSPEED.\$tcdelay_mon

Minimum: -5000.0 Default: 200.0 Maximum: 5000.0
 KCL/Data: RO Program: RO GET/SET_VAR: RO Data Type: REAL

Name: TCP Speed Prediction Delay Time Monitor (milliseconds)

Description: This is a copy of the TCPP equipment delay time (\$TCPPIR.\$tcdelay) which is currently being used as the prediction time within the TCPP softpart. This value is presented within this structure to provide a convenient monitor point for the user. This variable is intended to be used as a reference to indicate to the application processing task which value of TCDELAY is being used. Refer to the \$tcdelay field of the \$TCPPIR system variable to set the prediction equipment delay time.

Power Up: This value initializes to 0 at each power cycle and is updated automatically by the TCPP task softpart at every TCPP cycle.

User Interface Location: SYSTEM Variables screen

See Also: \$TCPPIR.\$tcdelay

\$TCPPSPEED.\$timestamp

Minimum: 0.0 Default: 0.0 Maximum: Maximum signed INTEGER
 KCL/Data: RO Program: RO GET/SET_VAR: RO Data Type: INTEGER

Name: TCP Speed Prediction Timestamp (ROS Ticks)

Description: This is the timestamp (in ROS ticks) corresponding to the most recent time when the current values of the \$TCPPSPEED data structure were computed and recorded.

Power Up: This value initializes to 0 at each power cycle and is updated automatically by the TCPP task softpart at every TCPP cycle.

User Interface Location: SYSTEM Variables screen

See Also: \$TCPPSPEED

\$TCPPSPEED.\$vspeed

Minimum: -2000.0 Default: 0.0 Maximum: 2000.0
KCL/Data: RO Program: RO GET/SET_VAR: RO Data Type: VECTOR of REAL

Name: TCP Speed Prediction Velocity Vector (millimeters / second)

Description: This system variable is the predicted speed of the TCP recorded as a vector with components x, y, and z. It is referenced to the WORLD frame for normal TCP operation, and to the currently selected Remote TCP frame for Remote TCP operation.

Power Up: These values initialize to 0.0 at each power cycle and are updated automatically by the TCPP task softpart at every TCPP cycle.

User Interface Location: SYSTEM Variables screen

\$TCPP_CFG STRUCTURE

Name: TCP Speed Prediction Configuration System Variable Structure

Description: This set of variables controls the mode of operation of TCP Speed Prediction. The individual fields within this structure are defined below.

User Interface Location: SYSTEM Variables screen

\$TCPP_CFG.\$group_num

Minimum: 1 Default: 1 Maximum: Maximum number of motion groups
KCL/Data: RW Program: RW GET/SET_VAR: RW Data Type: INTEGER

Name: TCP Speed Prediction Motion Group Number

Description: This system variable sets the number of the motion group for which the TCP Speed Prediction softpart will predict TCP speed. Currently this value is restricted to group 1 because TCP Speed Prediction is available only for group 1 motion..

Power Up: This value initializes to group 1 and will maintain its value over subsequent power cycles.

User Interface Location: SYSTEM Variables screen

\$TCPP_CFG.\$otf_tim_enb

Minimum: FALSE Default: TRUE Maximum: TRUE
KCL/Data: RW Program: RW GET/SET_VAR: RW Data Type: BOOLEAN

Name: TCP Speed Prediction On-The-Fly delay time update enable.

Description: This system variable when true allows for on-the-fly updating of the TCPP delay time value (\$TCPPIR.\$tcdelay). Since on-the-fly changes to the TCPP delay time might result in predicted speed value discontinuities, this variable might be set to false which will prevent delay time changes until the robot has stopped moving at the end of a program.

Power Up: This value initializes to TRUE and maintains its previous value over subsequent power cycles.

User Interface Location: SYSTEM Variables screen

See Also: \$TCPPIR.\$tcdelay

\$TCPP_CFG.\$warning_enb

Minimum: FALSE Default: TRUE Maximum: TRUE
KCL/Data: RW Program: RW GET/SET_VAR: RW Data Type: BOOLEAN

Name: TCP Speed Prediction Warning Message Enable

Description: This system variable when true allows the posting of warning level error messages. These are warnings only, not faults, which you might want not to be displayed. Currently these messages include the following:

- TCPP-011 "Pred time skips first motion"
- TCPP-018 "Begin Error Mode at line:nn"
- TCPP-019 "Speed Ovr Mode at line:nn"

Power Up: This value initializes to TRUE and maintains its previous value over subsequent power cycles.

User Interface Location: SYSTEM Variables screen

10.23 PROGRAM TOOLBOX (OPTION)

The Program ToolBox option consists of the following features. If you have installed the Program ToolBox option software, you can enable and use any of the following Program ToolBox options:

- Cross car mirror
- UTOOL adjust
- Flip knuckle
- Limit set

This section describes how to use the optional Program ToolBox functions.

10.23.1 Cross Car Mirror (option)

The cross car mirror function allows you to mirror a taught path across a car body, without having to choose a mirror plane or define reference points. Use the cross car mirror function to create a duplicate program for robots that perform the same function on different sides of a car body.

Without the cross car mirror function, you must create two programs and teach all points on both sides of the car body. With the cross car mirror function, you create one program and teach the points for one side of the car, and then use the cross car mirror function to create the program automatically for the robot across the line.

The cross car mirror function performs a specific mirror image function, as follows:

1. The cross car mirror function mirrors the program you select through the X-Z plane. By default, this is the program currently selected using the SELECT menu. See Figure 10-96.
2. The mirrored output is written to a destination program you specify. By default, the destination program is named MIRROR.TP.

The mirrored destination program will have mirrored UTOOL and UFRAME definitions. These new values are written as remarks and included at the beginning of the destination program, as shown in Figure 10-97.

Figure 10-96. Cross Car Mirror Function

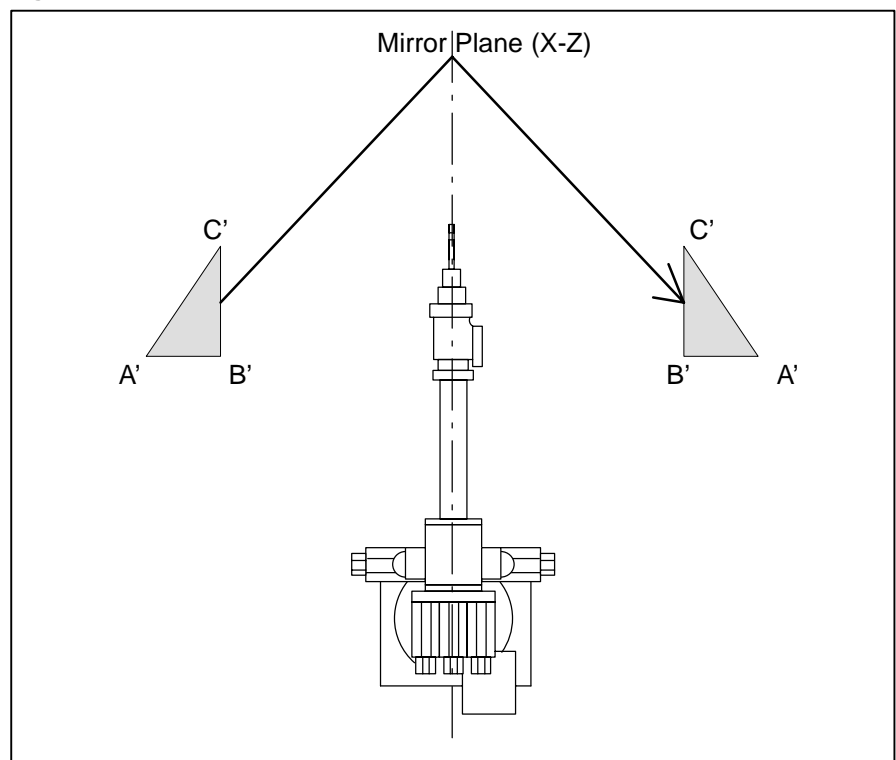


Figure 10-97. New UTOOL and UFRAME Values Displayed in Mirrored Destination Program

```

1.  !*****
2.  !MIRRORED UTOOL [1]
3.  !X =
4.  !Y =
5.  !Z =
6.  !W =
7.  !P =
8.  !R =
9.  !*****
10. !*****
11. !MIRROR UFRAME [1]
12. !X =
13. !Y =
14. !Z =
15. !W =
16. !P =
17. !R =
18. !*****
    
```

NOTE If UFRAME #0 was used for the entire program, the mirrored UFRAME value will not be written to the destination program.

Table 10-27 lists and describes the items you must set to use the cross car mirror function. Use Procedure 10-30 to perform program mirroring using the cross car mirror function.

Table 10-27. Cross Car Mirror Setup Items

ITEM	DESCRIPTION
Source Program	Source program is the name of the program for which you want to perform the cross car mirror function.
Destination Program	Destination program is the name of the program that will contain the program that results from performing the cross car mirror function on the original program.

Procedure 10-30 Performing the Cross Car Mirror Function

- Step**
- 1 Press MENUS.
 - 2 Select UTILITIES.
 - 3 Press F1, [TYPE].
 - 4 Select Prog ToolBox. You will see a screen similar to the following.

```

Prog ToolBox                               JOINT  50 %
                                           1 / 4
1  Cross Car Mirror
2  Utool Adjust
3  Flip Knuckle
4  Limit Set

[ TYPE ]                                     HELP

```

- 5 Move the cursor to Cross Car Mirror and press ENTER. You will see a screen similar to the following.

```

Prog ToolBox                               JOINT  50 %
                                           1 / 2
Cross Car Mirror
1  Source Program:                         STYLE ]
2  Destination Program:                    [ MIRROR ]

                                0% of program done

[ TYPE ]                               EXECUTE  [ CHOICE ]  HELP

```

- 6 Move the cursor to Original Program and press F4, [CHOICE]. Select the name of the program you want to mirror and press ENTER.
- 7 **If you want to use a name other than MIRROR.TP for the Destination Program**, move the cursor to Destination Program and press ENTER. Type the name you want the resulting mirrored program to have and press ENTER.
- 8 **To begin the mirroring**, press F3, EXECUTE.

NOTE If the destination program exists, you are asked whether to continue. If you select YES, the existing program will be overwritten. If you do not want to overwrite the program, select NO and enter a different program name.

The percentage of mirroring complete is displayed on the screen:

X% of program done

```
1 Set Device
2 Format
```

```
1 Floppy disk
2 Serial Printer
3 FROM Disk (FR:)
4 FTP (C1:)
5 Memory Card (MC:)
```

```
Continue displaying?
                     YES  NO
```

```
Press any key to exit
```

9 If an error occurs during the mirroring process, display the file **FR:MIRROR.DT** for detailed error information:

- a Press MENU.
- b Select FILE.
- c Press F1, [TYPE].
- d Select File.
- e Press F5, [UTIL].
- f Select Set Device.
- g Move the cursor to FROM Disk (FR:) and press ENTER.
- h Press F2, [DIR].
- i Select *.* and press ENTER.
- j Move your cursor to MIRROR.DT.
- k Press NEXT, >, and press F3, DISPLAY. The file will be displayed on the screen.
- l To continue displaying, press F4, YES, otherwise press F5, NO.
- m When the file is finished being displayed, press any key to continue.

10 If power is interrupted while a program is being mirrored, do the following:

- a Restore power.
- b Delete the specified destination program (MIRROR.TP, by default).
- c Repeat the cross car mirror function procedure.

10.23.2 UTOOL Adjust (option)

The UTOOL adjust function allows you to correct for an invalid or missing UTOOL definition, without changing the physical positions to which the robot moves in existing programs and position registers. The UTOOL adjust function is most frequently used to define a tool frame for a robot that might have been taught originally without a tool definition, or to change a tool definition that was taught using the three-point method to one taught using the six-point method.

The UTOOL adjust feature allows you to correct an incorrect UTOOL definition without changing an already taught program. You do this by specifying the number of the UTOOL you want to adjust, the new UTOOL, and which programs to adjust with the new UTOOL.



WARNING

Using the UTOOL Adjust function will change the source UTOOL to the new values you enter. If you choose to adjust only one program, then any other programs on the controller will not run correctly because they will use UTOOL values that are different than the ones with which they were taught. Be aware of the effect of the UTOOL you are changing and the effect it will have on other programs before you use the UTOOL Adjust function; otherwise, you could injure personnel or damage equipment.

You can specify the correct UTOOL using one of the following methods:

- **By Number** – Specify the number of a UTOOL you have defined previously using the tool frame setup procedures in Section 3.8.1.
- **Direct Entry** – Enter the values of the tool frame X, Y, Z, W, P, and R on the Utool Adjust screen. You must have determined these values previously, either through measurement or calculation.

Table 10–28 lists and describes the items you must set to use the UTOOL adjust function. Use Procedure 10–31 to adjust a UTOOL.

Table 10–28. UTOOL Adjust Setup Items

ITEM	DESCRIPTION
Source Utool	This is the UTOOL you want to adjust. You can choose from UTOOL 1 through 5.
Position Register Adjustment	Pos Reg Adjustment allows you to choose which position registers to adjust, as follows: <ul style="list-style-type: none"> • HOME – Adjust only the home position register • ALL – Adjust all position registers on the controller • NONE – Do not adjust any position registers

Table 10-28. (Cont'd) UTOOL Adjust Setup Items

ITEM	DESCRIPTION
Adjustment Type	Adjustment Type allows you to specify whether to adjust all programs on the controller, or only one. <ul style="list-style-type: none"> • ALL – Adjust all programs on the controller • ONE – Adjust only one program
Program to Adjust	If the Adjustment Type is set to ONE, Program to Adjust is the name of the program to which you want to apply UTOOL adjustments.

Procedure 10-31 Adjusting a UTOOL

**WARNING**

Using the UTOOL Adjust function will change the source UTOOL to the new values you enter. If you choose to adjust only one program, then any other programs on the controller will not run correctly because they will use UTOOL values that are different than the ones with which they were taught. Be aware of the effect of the UTOOL you are changing and the effect it will have on other programs before you use the UTOOL Adjust function; otherwise, you could injure personnel or damage equipment.

- Step**
- 1 Press MENUS.
 - 2 Select UTILITIES.
 - 3 Press F1, [TYPE].
 - 4 Select Prog ToolBox. You will see a screen similar to the following.

Prog ToolBox	JOINT 50 %
	2/4
1 Cross Car Mirror	
2 Utool Adjust	
3 Flip Knuckle	
4 Limit Set	
[TYPE]	HELP

- 5 Move the cursor to Utool Adjust and press ENTER. You will see a screen similar to the following.

Prog ToolBox		JOINT	50 %
Utool Adjust			1 / 4
1	Source Utool:	1	
2	Pos Reg Adjustment:	HOME	
3	Adjustment type:	ALL	
4	Program to Adjust:	*****	
[TYPE]	METHOD		HELP

- 6 Move the cursor to each of the items on the screen and set them as desired.

NOTE DispenseTool does not use the position register adjustment.

- 7 Press F3, METHOD. You will see a screen similar to the following.

Prog ToolBox		JOINT	50 %
Utool Adjust			1 / 2
1.	By Number		
2.	Direct Entry		
Please choose method:			
[TYPE]			HELP

- 8 Move the cursor to the method you want to use and press ENTER.
- For **By Number**, see the following screen for an example.

Prog Adjust		JOINT	50 %
Utool Adjust			1 / 1
Correct Utool Number:		5	
0% of programs done			
[TYPE]	EXECUTE		HELP

- a Type the number of the UTOOL you want to use.

Specify a UTOOL number that contains the correct tool definition values. By default, the specified UTOOL number is 5. You can use the tool frame setup procedures in Section 3.8.1 to define the frame you want and specify the number of that frame here.

- b Press ENTER.

- For **Direct Entry**, see the following screen for an example.

Prog ToolBox		JOINT 50 %
Utool Adjust		1/6
Correct Utool		0.000
1	X:	0.000
2	Y:	0.000
3	Z:	0.000
4	W:	0.000
5	P:	0.000
6	R:	0.000
0% of programs done		
[TYPE]	EXECUTE	HELP

- Move the cursor to each item and type the numeric values.
- Press ENTER after each entry.

9 To adjust the UTOOL, press F3, EXECUTE.

The following actions will occur:

- The time and date of the UTOOL adjustment is written to the file **FR:UTADJUST.DT**.
- A backup of the original program is made to **UTBACK.TP**.
- The current program is adjusted.
- When finished adjusting the current program, the backup in **UTBACK.TP** is cleared, and a message is written to **FR:UTADJUST.DT** stating that the change is complete for that program.
- The next program is adjusted, if the Adjustment Type is ALL, until all programs have been adjusted. The status of the adjusted programs is written to **FR:UTADJUST.DT**.
- The selected position registers are adjusted.
- The position registers adjusted, or none, are recorded in **UTADJUST.DT**.

The percentage of programs adjusted is displayed on the screen:

x% of program done

10 If power is interrupted while a program is being adjusted, do the following:

- Check **FR:UTADJUST.DT** to determine which programs have been adjusted successfully:
 - Press **MENUS**.
 - Select **FILE**.
 - Press **F1, [TYPE]**.

```
1 Set Device
2 Format
```

```
1 Floppy disk
2 Serial Printer
3 FROM Disk (FR:)
4 FTP (C1:)
5 Memory Card (MC:)
```

```
Continue displaying?
                    YES  NO
```

```
Press any key to exit
```

- (4) Select File.
 - (5) Press F5, [UTIL].
 - (6) Select Set Device.
 - (7) Move the cursor to FROM Disk (FR:) and press ENTER.
 - (8) Press F2, [DIR].
 - (9) Select *.* and press ENTER.
 - (10) Move your cursor to UTADJUST.DT.
 - (11) Press NEXT, >, and press F3, DISPLAY. The file will be displayed on the screen.
 - (12) To continue displaying, press F4, YES, otherwise press F5, NO.
 - (13) When the file is finished being displayed, press any key to continue.
- b** Use the contents of **UTBACK.TP** to restore the program that was being adjusted when the error occurred. You do this by copying **UTBACK.TP** to the program you were adjusting. Refer to Section 9.2.4 for information on copying a program.
- c** Continue adjusting the programs that were not adjusted. Repeat this procedure, set Adjustment Type to ONE, specify the name of the program to adjust, and complete the procedure. Do this for each of the programs you must adjust.

10.23.3 Flip Knuckle (Option)

The flip knuckle function allows you to rotate the knuckle configuration (wrist) in a teach pendant program. You can use the flip knuckle function to rotate the wider side of the robot's wrist 180 degrees, in order to prevent the hoses and cables from rubbing on it. When the cables run across the narrower side of the wrist, there is less friction between the cables and the wrist when the robot moves. See Figure 10-98.

Without the flip knuckle function, you must re-teach the robot programs to re-orient the wrist if the cables rub on the wrist. The flip knuckle function changes the orientation of the wrist for all points in a program automatically to accommodate the rotated wrist.

The flip knuckle function is often used after using the cross car mirror function, so that the mirrored teach pendant program can better duplicate the original teach pendant program and prevent hoses and cables from rubbing.

When the joints are flipped, all of the positions in the program you specify as the source program are flipped. You select how you want to flip the joints using the Flip Type item on the Flip Knuckle screen.

If flipping some of the positions in the program causes a joint error, you will have to either re-teach those positions so that they do not cause joint errors, or select another Flip Type and try again.

Figure 10-98. Flip Knuckle Function

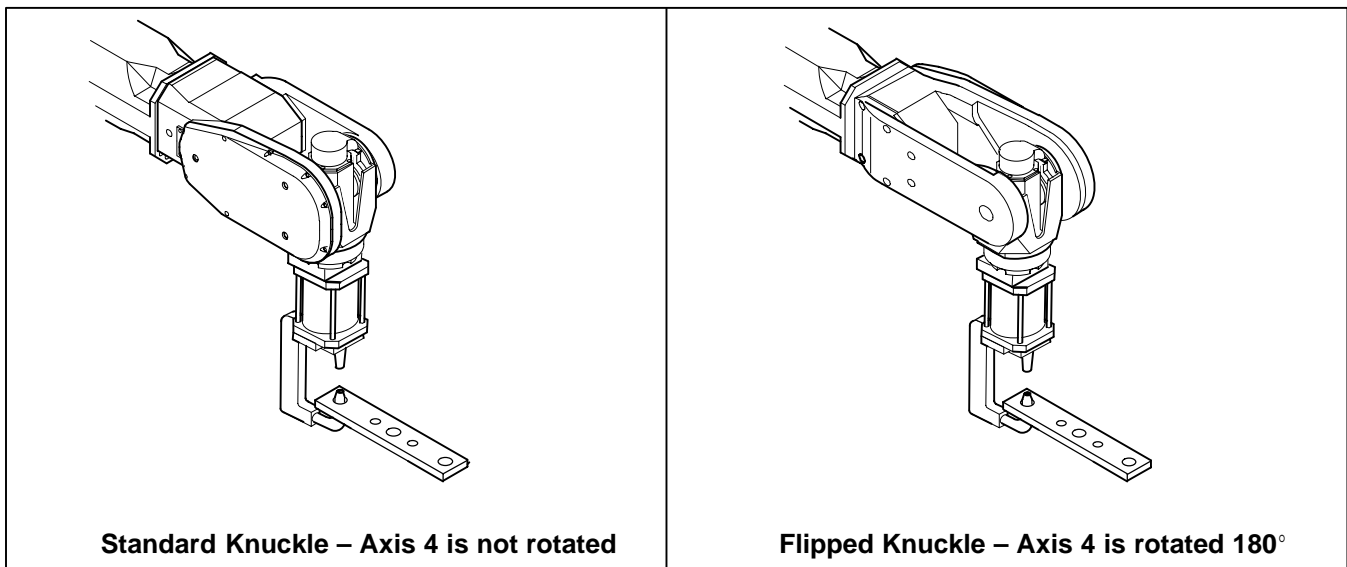


Table 10-29 lists and describes the items you must set to use the flip knuckle option. Use Procedure 10-32 to perform the flip knuckle function.

Table 10-29. Flip Knuckle Setup Items

ITEM	DESCRIPTION
Source Program	Source program is the name of the program on which you want to perform the flip knuckle function.
Destination Program	Destination program is the name of the program that will contain the program that results from performing the flip knuckle function on the source program.
Flip Type	Flip type is the way you want to flip the knuckle: <ul style="list-style-type: none"> • Type 1 (+ -): J4 = +180 J6 = -180 • Type 2 (- +): J4 = -180 J6 = +180 • Type 3 (+ +): J4 = +180 J6 = +180 • Type 4 (- -): J4 = -180 J6 = -180

Procedure 10-32 Performing the Flip Knuckle Function

- Step**
- 1 Press MENUS.
 - 2 Select UTILITIES.
 - 3 Press F1, [TYPE].
 - 4 Select Prog ToolBox. You will see a screen similar to the following.

```

Prog ToolBox                               JOINT  50 %
                                           3 / 4

 1  Cross Car Mirror
 2  Utool Adjust
 3  Flip Knuckle
 4  Limit Set

[ TYPE ]                                   HELP

```

- 5 Move the cursor to Flip Knuckle and press ENTER. You will see a screen similar to the following.

```

Prog ToolBox                               JOINT  50 %
Flip Knuckle                               1 / 3
 1  Source Program:                         STYLE
 2  Destination Program:                    FLIP
 3  Flip Type:                               Type1(+ -)

                                0% of program done

[ TYPE ]           EXECUTE   [ CHOICE ]   HELP

```

- 6 Move the cursor to Original Program and press F4, [CHOICE]. Select the name of the program you want and press ENTER.

```

1 Type 1(+ -):J4=+180 J6=-180
2 Type 2(- +):J4=-180 J6=+180
3 Type 3(+ +):J4=+180 J6=+180
4 Type 4(- -):J4=-180 J6=-180

```

- 7 Move the cursor to Destination Program and press ENTER. Type the name of the program you want the resulting flipped program to have and press ENTER.
- 8 Move the cursor to Flip Type, press F4, [CHOICE], and select the flip type you want to use. Refer to Table 10-29.
- 9 **To begin the flipping**, press F3, EXECUTE.

NOTE If the destination program exists, you are asked whether to continue. If you select YES, the existing program will be overwritten. If you do not want to overwrite the program, select NO and enter a different program name.

The percentage of flipping complete is displayed on the screen:

X% of program done

Continue?

Yes No

NOTE If any points could not be flipped due to joint limits encountered, you will be notified and prompted whether to continue.

- If you choose to continue, the destination program will be written with the points that caused the joint limits. These points are written to the file **FR:FLIP.DT**. If you try to run the destination program, you will encounter limit errors on the points that caused the joint limits. You will not be able to move to these points until you have touched up or retightened them so limit errors will not occur.
- If you choose not to continue, the destination program will not be written and you can choose another flip type and try again.

- 10 **If power is interrupted** while a program is being flipped, do the following:
 - a Restore power.
 - b Delete the specified destination program (FLIP.TP, by default).
 - c Repeat the procedure.

10.23.4 Limit Set (option)

The limit set function allows you to determine the software limits for a robot automatically, and to determine the axis 1 hard stop locations. If you want to set the axis limits without the limit set function, you must run all programs step-by-step, record the maximum value for each angle, and then set the axis joint limits manually. Then, you would have to jog the robot to the axis 1 joint limits and determine the best location for the axis 1 hard stops manually.

The limit set function 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 takes this information and sets the specified joint limits automatically, while taking into account a limit buffer you specify. The limit set function also reports the appropriate locations for the axis 1 hard stops, based on the same maximum taught joint angles in all programs.

By default, software axis limits will be set only for axes 1, 2, and 3. You can specify that they be set for more axes, if desired.

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

You can set a limits buffer of from 0 to 50 degrees. The limits buffer is set to 10 degrees, by default. A limits buffer of 10 degrees provides adequate ease in many cases. After using the limit set function, if you encounter many joint limit errors while running programs, increase the limits buffer and try again.

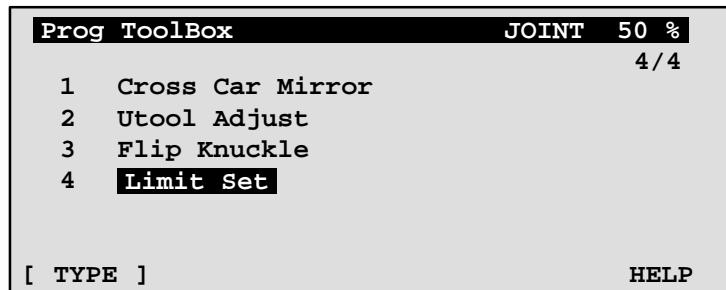
Before you set limits, you select the axes for which you want to set limits and change the value of the limits buffer, if desired.

You can reset the limits to factory settings if you do not want to use the new ones.

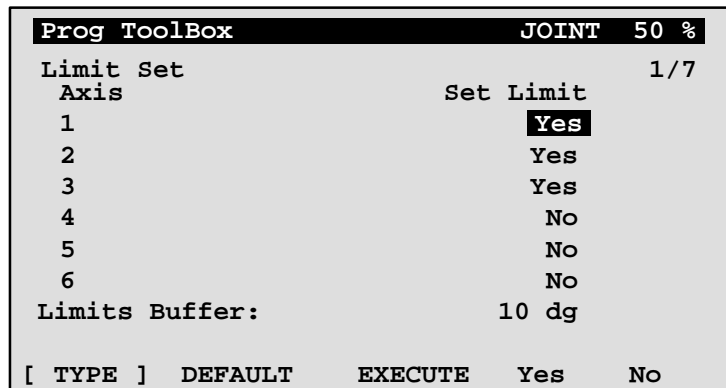
Use Procedure 10-33 to perform the limit setting function.

Procedure 10-33 Using the Limit Set Function to Set Software Axis Limits

- Step**
- 1 Press **MENUS**.
 - 2 Select **UTILITIES**.
 - 3 Press **F1**, [**TYPE**].
 - 4 Select **Prog ToolBox**. You will see a screen similar to the following.



- 5 Move the cursor to **Limit Set** and press **ENTER**. You will see a screen similar to the following.



- 6 **To select an axis for limit setting**
 - a Move the cursor to the number of the axis you want to select.
 - b Press **F4**, **Yes**, to select the axis to set. If you press **F5**, **No**, the limit will not be set for that axis.
- 7 To set the limits buffer, move the cursor to **limits buffer**, type the number of degrees, and press **ENTER**.

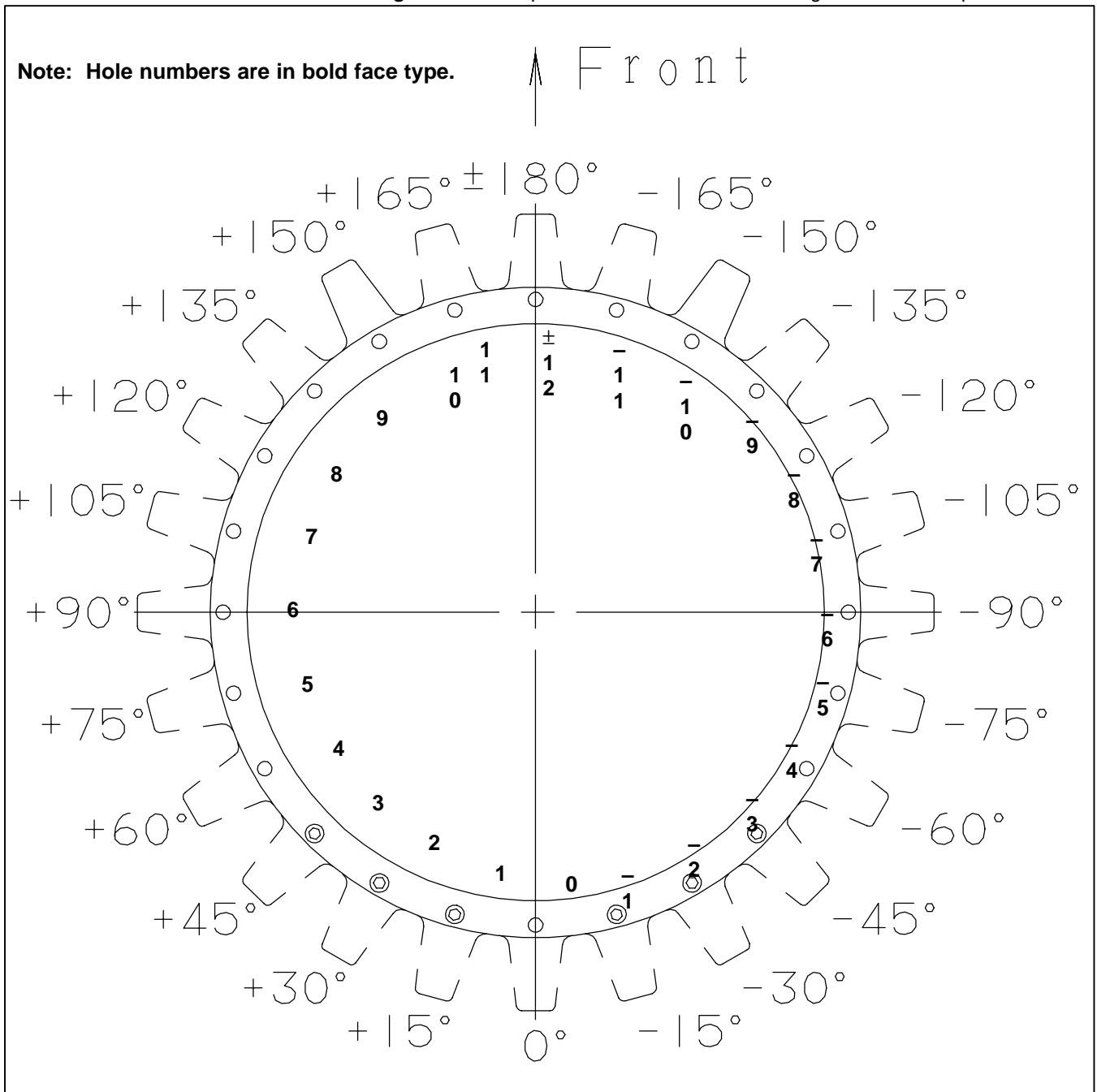
8 To set the axis limits, press F3, EXECUTE.

When the limits have been set, you will see a screen similar to the following.

Prog ToolBox		JOINT 50 %	
New Limits			
Axis	LOWER	UPPER	
J1	-56	78	dg
J2	-23	30	dg
J3	-22	25	dg
J4	-190	200	dg
J5	-10	90	dg
J6	-100	280	dg
Min Hole # for Axis 1 Hardstop: -4			
Max Hole # for Axis 1 Hardstop: 6			
YOU MUST COLD START TO TAKE EFFECT			

The hole numbers displayed are for axis 1 hard stops. The center hole of the physical hard stop must align with the reported axis 1 hard stop hole. See Figure 10-99.

Figure 10-99. Top view of Robot Base for Locating Axis 1 Hard Stop Location



NOTE The reported hole number refers to the location for the center bolt hole of the physical hard stop.

NOTE The following operation is not supported on some robot models.

9 To reset the axis limits to their factory settings, press F2, DEFAULT.

NOTE You must always cold start the controller in order for software axis limit changes to take effect.

10 When you are finished setting axis limits, cold start the controller so that the new software axis limits can take effect:

- a** If the controller is ON, turn it OFF.
- b** On the teach pendant, press and hold the SHIFT and RESET keys.
- c** While still pressing the teach pendant keys, press the ON button on the operator panel or operator box.
- d** After the a teach pendant screen has been displayed, release the teach pendant keys.

NOTE If you encounter many joint limit errors while running programs, increase the limits buffer and try running the programs again.

11 FANUC SENSOR INTERFACE

11

FANUC SENSOR INTERFACE

Topics In This Chapter		Page
Communication Protocol	The FANUC Sensor Interface has a proprietary software flow control mechanism and uses its own data format.	11-2
	• Flow Control Mechanism	11-2
	• Data Format	11-2
	• Hand Shaking	11-4
Programming	Your program must contain the program elements that control receiving and transforming the position offset data	11-6
Sensor Setup and Hardware Connections	To use an external sensor device with the R-J2 controller, you must setup the RS-232 port, and connect the RS-232 port to an external sensor.	11-7

The FANUC Sensor Interface is an optional feature that enables the R-J2 controller to communicate with an external sensor device through the RS-232-C serial port. The sensor device provides the R-J2 controller with position offset data. The R-J2 controller uses the position offset data for robot motion compensation. This option is required to interface application tools to the V-120 Vision System.

Usually, a sensor has a coordinate system that differs from the coordinate system used by the robot. Because of this difference, the offset data must be transformed so the result reflects a valid offset for the coordinate system of the robot.

After the R-J2 controller receives the offset data, you can use the offset data to compensate any taught position in your program. See Section 11.2 for more information about programming.

11.1 COMMUNICATION PROTOCOL

The FANUC Sensor Interface has a proprietary software flow control mechanism and uses its own data format. This communication protocol is similar to Xmodem protocol, but uses a very simple control sequence. It has a fixed data format and is equipped with a block checksum character for error detection.

11.1.1 Flow Control Mechanism

Four different flow control codes are available:

- ENQ
- ACK
- NAK
- EOT

ENQ (enquiry)
0x05 in hex representation

For every transmitted packet, the sending end sends an ENQ request to the receiving end to begin the data exchange. The sending end waits for the receiving end to respond with an ACK signal. If the sending end does not receive any acknowledgement from the receiving end within a predefined period of time, an error occurs.

ACK (acknowledgement)
0x06 in hex representation

The receiving end sends ACK when it is ready for data transmission. If the receiving end receives an ENQ but is not ready for data exchange, it sends a NAK.

NAK (not acknowledged)
0x95 in hex representation

The receiving end sends a NAK if it is not ready for data exchange when an ENQ arrives.

EOT (end of transmission)
0x84 in hex representation

After the sending end sends all the data in a packet, it notifies the receiving end that the packet sending session is finished by sending an EOT.

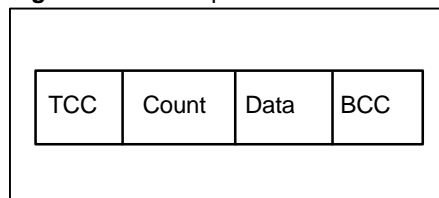
11.1.2 Data Format

The FANUC Sensor Interface recognizes five different fixed length packet formats:

- 0X55
- 0X5A
- 0X36
- 0X3A
- 0X6C

These packet formats are composed as shown in Figure 11-1 and are detailed in Table 11-1.

Figure 11-1. Composition of FANUC Sensor Data Packet



TCC
Transfer Control Character
 1 Byte

The receiving end uses the TCC to interpret the corresponding data in the packet.

Count
 1 Byte

The Count is the number of bytes that follows in the Data section. (BCC is not included.)

Data
 Multiple Bytes

The Data section represents the data to be exchanged. The Data section allows multiple bytes, but the number of bytes is fixed for each TCC.

BCC
Block Checksum Character
 1 Byte

The BCC is computed by taking Exclusive-Or (XOR) of all bytes in the Count and data section. This is an even parity checking mechanism for error detection.

Table 11-1. FANUC Sensor Data Formats

Function	TCC	Type	Count	Data	Unit												
Send the data contained in a register to the sensor	0X55	Output	3	Byte 1 to 3 is integer data.	None												
Receive data for a register.	0X5A	Input	4	Byte 1 is the register number. Bytes 2 to 4 are data for the register.	None												
Receive data for a position register	0X36	Input	12	Byte 1 is the position register number. (PR[1] – PR[5]) Byte 2 to 4 is the x component. Byte 5 to 7 is the y component. Byte 8 to 10 is the z component. Byte 11 and 12 is the orientation. Orientation is not used.	The unit for the x, y, and z components is 0.01 mm and the unit for rotation is 0.01 deg. For example, X=0X00012C is equal to 4 mm.												
Receive 3 point data.	0X3A	Input	28	Byte 1 indicates whether the data is the taught position or is the offset data. Byte 1-0: taught position Byte 1-1 offset data Byte 2 to 28 contain position data for the x, y, z components of three positions.	0.01mm												
Receive transformation matrix	0X6C	Input	36	Contains the elements of the homogenous transformation matrix: <table style="margin-left: 20px; border-collapse: collapse;"> <tr> <td>a11</td> <td>a12</td> <td>a13</td> <td>a14</td> </tr> <tr> <td>a21</td> <td>a22</td> <td>a23</td> <td>a24</td> </tr> <tr> <td>a31</td> <td>a32</td> <td>a33</td> <td>a34</td> </tr> </table> Each element occupies 3 bytes. The order of transmission is a11, a21, a31, ..., a14, a24, a34.	a11	a12	a13	a14	a21	a22	a23	a24	a31	a32	a33	a34	a*1, a*2, a*3 (*=1,2,3): 0.00001 a*4 (*=1,2,3) : 0.01 mm
a11	a12	a13	a14														
a21	a22	a23	a24														
a31	a32	a33	a34														

11.1.3 Hand Shaking

There are five handshaking sequences for each packet:

- Send the contents of a register to the sensor.
- Receive a register number and its data.
- Receive data for a position register.
- Receive three point data.
- Receive the transformation matrix.

Figure 11-2 through Figure 11-6 illustrate each of these handshaking sequences.

Figure 11-2. Handshaking When the Contents of a Register is Sent to the Sensor

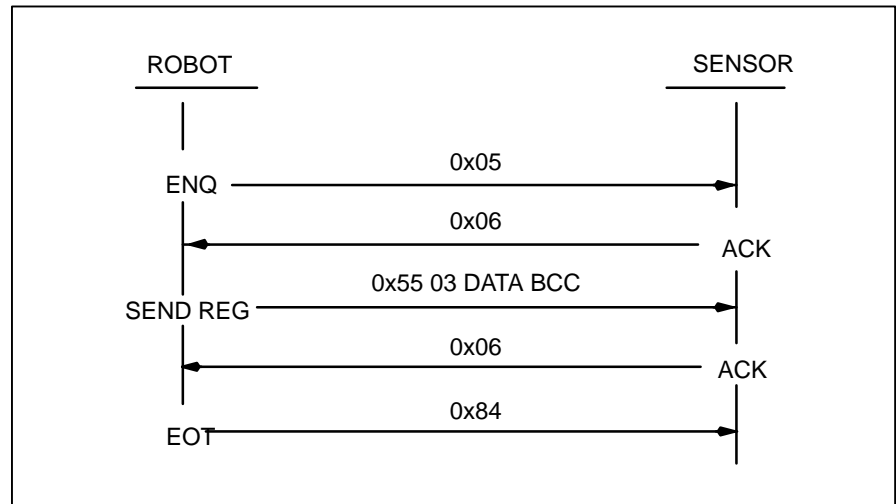


Figure 11-3. Handshaking When a Register Number and Its Data is Received

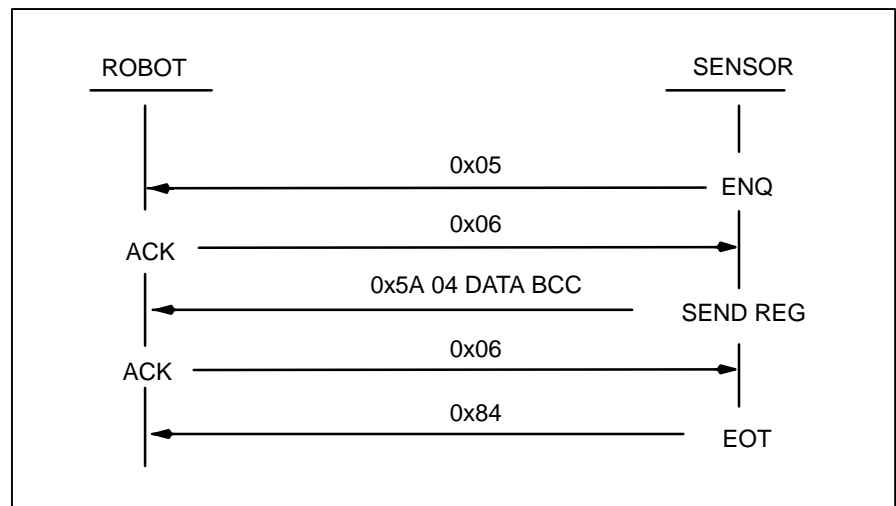


Figure 11-4. Handshaking When Position Register Data is Received

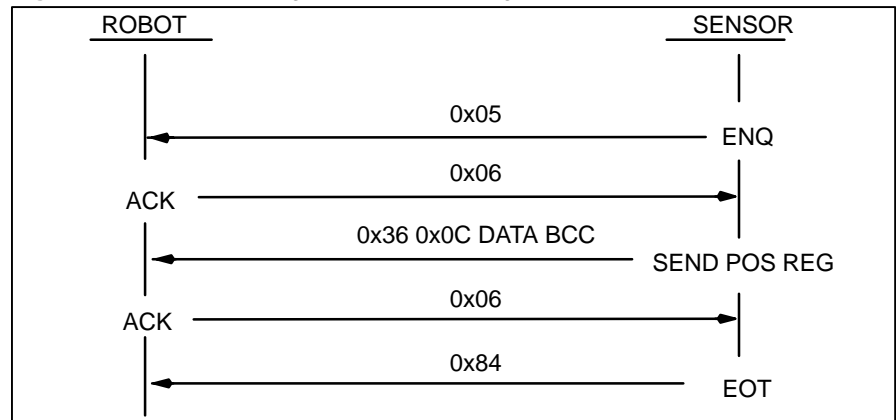
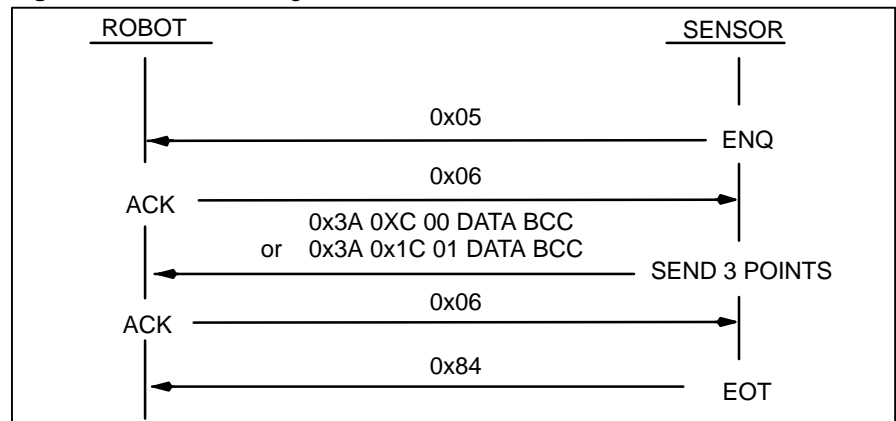
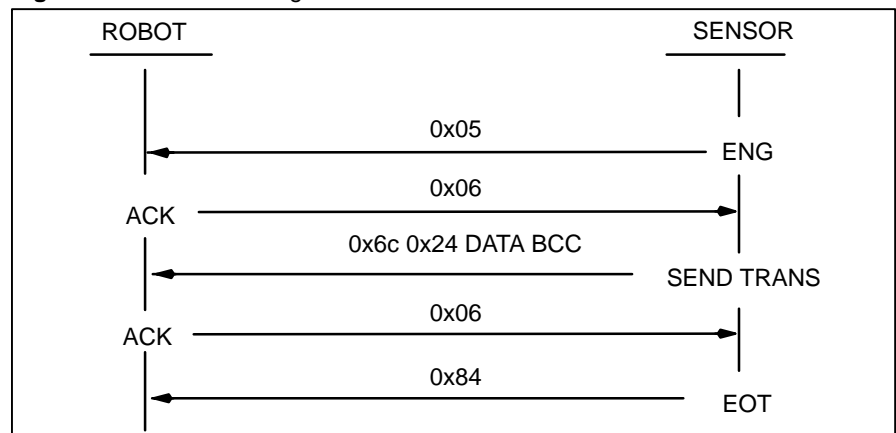


Figure 11-5. Handshaking When Three Point Data is Received



The three point data transmitted from the sensor (V-120 vision system) is set in the position registers specified by the system variables \$SENS_IF[].

Figure 11-6. Handshaking When the Transformation Matrix is Received



The transmitted transformation matrix from the sensor is set in the position register specified by the \$SENS_IF[] system variables. Refer to Section 8.6 and the *SYSTEM R-J2 Controller Software Reference Manual* for more information on system variables.

11.2 PROGRAMMING

There are four program instructions associated with the FANUC sensor interface:

- SEND R[*]
- RCV R[*], LBL[*]
- RCV R[*] LBL[x] TIMEOUT, LBL[y]
- CALMATRIX

SEND R[*] sends out a packet with TCC = 0X5A. RCV R[*], LBL[*] waits for input until the register is set, then continues to the OFFSET condition instruction. If a communication error occurs, the TPE will jump to LBL[1]. CALMATRIX computes the transformation matrix for offset data.

The OFFSET program element can then be used in your program to compensate the motion of any recorded robot position. Refer Chapter 6 for more information about program instructions.

In the programming example shown in Figure 11-7, sensor will send all the offset data to the position registers. At the end, sensor set R[2] to 1 so TPE will continue to execute the program. At that point, all the P[1], P[2] and P[3] will be offset by the sensor input.

CAUTION

Recorded positions are not affected by UFRAME and UFRAME has no affect during playback. However, **position registers are recorded with respect to UFRAME**. If you change UFRAME, any recorded position registers will also change.

Figure 11-7. FANUC Sensor Interface Program Example

```

1: R[1] = 2
2: R[2] = 0
3: !Notify sensor to use R[2] to terminate comm
4: SEND R[1]
5: !Get the input, at the end set R[2] to 1
6: RCV R[2], LBL[1]
7: OFFSET CONDITION PR[1]
8: J P[1] 100% FINE OFFSET
9: J P[2] 100% FINE OFFSET
10: J P[3] 100% FINE OFFSET

```

11.3 SENSOR SETUP AND HARDWARE CONNECTIONS

To use an external sensor device with the R-J2 controller, you must:

- Set up the RS-232 port.
- Connect the RS-232 port to an external sensor.

You set up the port using the Port Init screen in the SETUP menu. See Section 9.1.1. The external sensor device uses the following default communications setting:

- 4800 baud
- 1 stop bit
- Odd parity

Refer to Section 9.1.1 for information on how to set up a port.

You use a NULL modem cable to connect the RS-232 port to a sensor device. See Table 11-2 for the pin connector layout.

Table 11-2. NULL Modem Cable Pin Connector Layout

RS-232 Port	External Sensor
Pin 2 (TXD)	Pin 3 (RXD)
Pin 3 (RXD)	Pin 2 (TXD)
Pin 4 (RTS)	Pin 5 (CTS)
Pin 5 (CTS)	Pin 4 (RTS)
Pin 6 (DSR)	Pin 20 (DTR)
Pin 20 (DTR)	Pin 6 (RSD)

12

AUTOMATIC TOOL CENTER POINT

MARO2HT4405801E

12-1

Topics In This Chapter		Page
Overview	Auto TCP allows you to re-teach the the Tool Center Point quickly and consistently after a tool has been replaced.	12-2
Hardware Requirements and Installation	A string sensor and Tool Center Point attachment device are required. <ul style="list-style-type: none">● Installing the String Sensor and TCP Attachment Device 12-3● Sensor I/O Signal Cables 12-5● Sensor Setup 12-7● Testing Data Ports 12-10	12-3
TCP Orientation Setup	The TCP Orientation Menu can set up the TCP's w, p, and r values.	12-11
Auto Setup	The Auto Setup menu allows you to set up the TCP's x, y, and z axis values ... <ul style="list-style-type: none">● Tool Orientation 12-13● Automatically Generating Positions 12-14● Manually Defining Positions 12-14● Automatic and Manual Position Generation 12-14	12-13
Automatically Calculating a new UTOOL TCP	You can use Auto TCP to calculate a new UTOOL TCP Automatically	12-20

12.1 OVERVIEW

The Automatic Tool Center Point (Auto TCP) feature allows you to define a Tool Center Point (TCP) quickly and accurately. Because two different operators might use different methods to define a TCP, final TCP values might differ by as much as 20 mm. Auto TCP is designed to prevent that degree of difference in the final TCP values.

Auto TCP generates the TCP through the use of data provided by a sensor that is physically connected to the TCP, and mounted to a fixture in the workcell. After Auto TCP has been set up with the required data, a new TCP can be calculated in a much shorter amount of time. This can reduce the amount of robot down time during a tooling change.

After Auto TCP has calculated a new TCP value, it can be assigned to any of the available UTOOL frames. In order for Auto TCP to calculate a new TCP, you must do the following:

- Install the string sensor and Auto TCP Attachment device. Refer to Procedure 12-1 .
- Connect and Configure the I/O ports. Refer to Procedure 12-6 .
- Define the start position and the pull position, in the Auto Setup menu. Refer to Procedure 12-8 .
- Run the Auto TCP program. Refer to Procedure 12-9 .

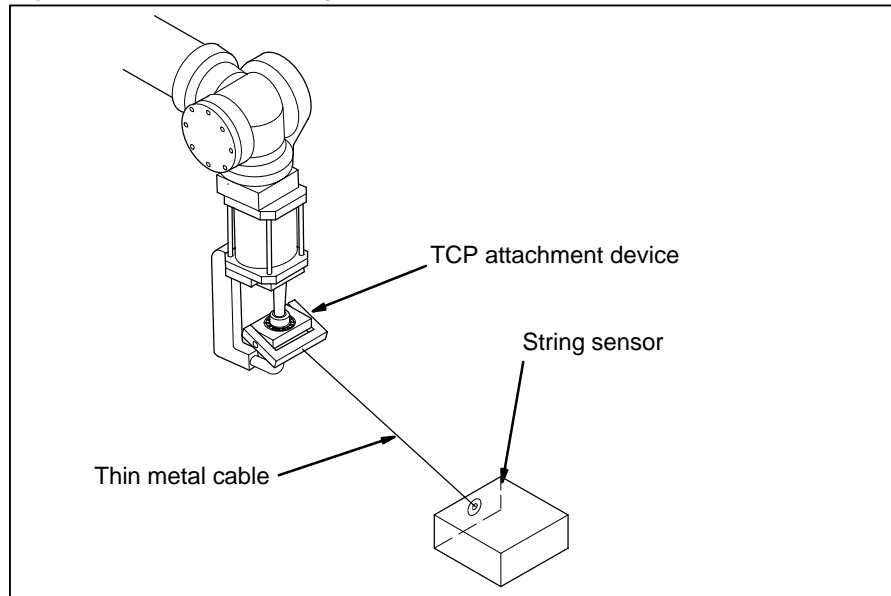
**WARNING**

Be sure to step through all programs that use automatically generated tool center points at a slow speed before you run them at production speed. Otherwise, you could injure personnel or damage equipment.

12.2 HARDWARE REQUIREMENTS AND INSTALLATION

Auto TCP requires a string sensor that must be mounted in the workcell, and a TCP attachment device. See Figure 12-1. The string sensor, which consists of a thin metal cable that is connected to a spool inside a case, has two I/O signal cables that must be connected to either controller DIN ports or RDI ports. The string sensor is the physical interface between the Auto TCP software and the robot itself.

Figure 12-1. Auto TCP String Sensor and TCP Attachment Device



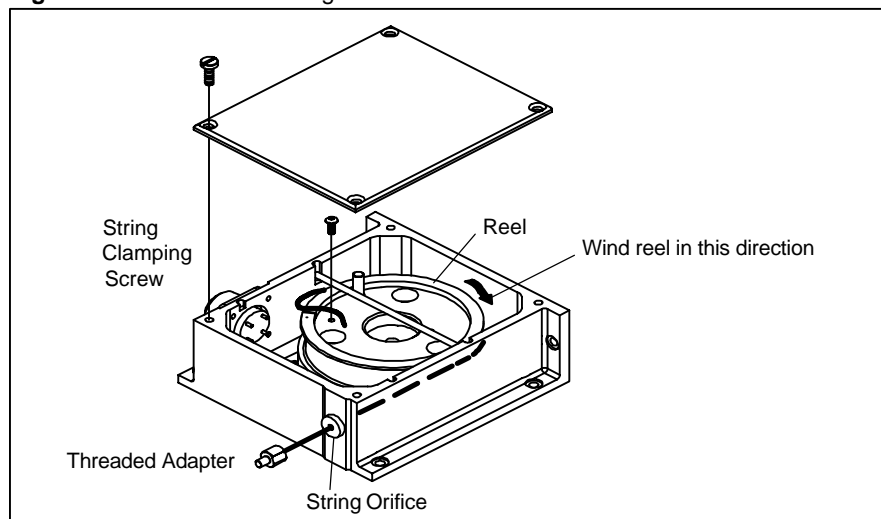
12.2.1 Installing the String Sensor and TCP Attachment Device

You must install the string sensor and Auto TCP attachment device in a location appropriate for your workcell. The details of installation will vary depending on your specific requirements. Figure 12-1 shows a typical installation of the string sensor and Auto TCP attachment device you can use as a guide to completing your installation. Procedure 12-1 is a general overview of the hardware setup.

Procedure 12-1 Setting Up the String Sensor and TCP Attachment Device

- 1 Secure the String Sensor to a fixed position in the workcell.
- 2 Place the Tool Center Point Attachment Device in the TCP.
- 3 Connect the two I/O signal cables to available I/O ports. Set up the ports as either RDI or DI ports. Refer to Procedure 12-3 and Procedure 12-4 .

Procedure 12-2 Replacing the Auto TCP String Sensor String

Figure 12-2. Auto TCP String Sensor


- 1 Remove the screws that secure the lid to the Auto TCP sensor.
- 2 Remove the lid from the Auto TCP sensor.
- 3 Loosen the string clamping screw and remove any remaining string inside the sensor.
- 4 Wind the reel clockwise at least 12 turns. See Figure 12-2.
- 5 When you have turned the reel a sufficient number of turns to be able to wind the entire length of the string, lock the reel in that position.
- 6 Thread the replacement string through the threaded adapter so the ball at the end of the string is inside the threaded opening.
- 7 Thread the replacement string through the string orifice on the Auto TCP sensor.
- 8 Feed the end of the replacement string through the slot in the reel, and secure the string with the clamping screw.
- 9 While keeping tension on the string, unlock the reel and allow the string to wind onto the reel slowly.


CAUTION

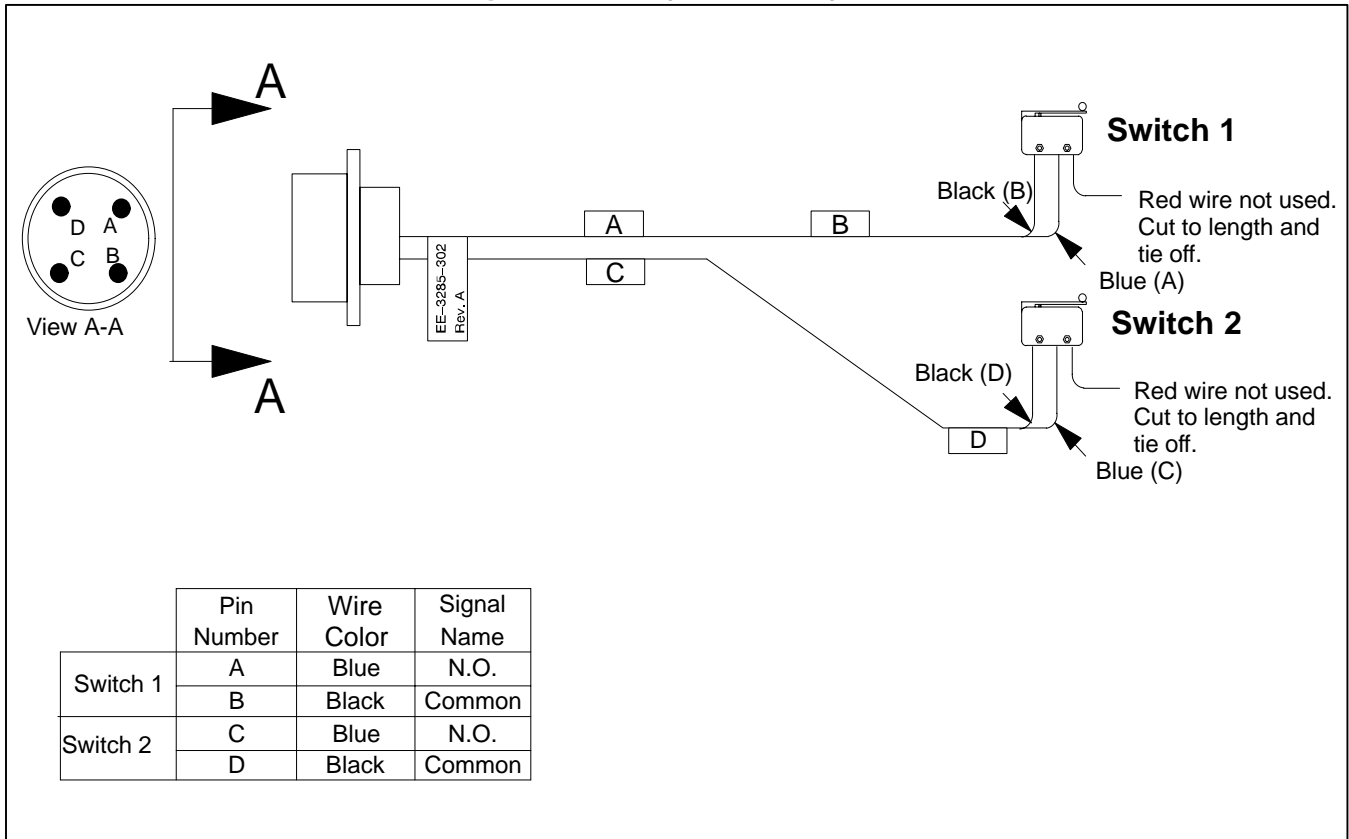
Do not release tension on the string and allow it to rapidly wind onto the reel. Otherwise, you will damage the replacement string.

- 10 Re-install the lid on the Auto TCP sensor.
- 11 Re-install the screws that secure the lid to the Auto TCP sensor.

12.2.2 Sensor I/O Signal Cables

You must connect the two I/O signal cables to either DIN ports or RDI ports on the controller, and configure the selected ports to recognize this connection. Figure 12-3 shows the I/O cables and connectors. Use Procedure 12-3 to install the I/O signal cables and Procedure 12-4 to configure I/O.

Figure 12-3. String Sensor I/O Signal Cables



Procedure 12-3 Installing Sensor I/O Signal Cables

- Step**
- 1** Select two available data ports on the controller.
 - 2** Connect the appropriate connector of the cable to the data port on the controller.
 - 3** Connect the other end of the cable to the String Sensor. Refer to Procedure 12-3 .

Procedure 12-4 Configuring the I/O Ports

- Condition**
- You have installed the String Sensor and TCP Attachment Device. Refer to Procedure 12-1 .
 - You have installed the String Sensor I/O cables. Refer to Procedure 12-3 .

NOTE The following steps describe configuring DI ports. Refer to Section 3.1 for more detailed information. RDI port configuration is similar.

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

I/O Digital In			JOINT	50 %
#	SIM	STATUS	1 / 256	
DI [1]	*	OFF []	
DI [2]	*	OFF []	
DI [3]	*	OFF []	
DI [4]	*	OFF []	
DI [5]	*	OFF []	
DI [6]	*	OFF []	
DI [7]	*	OFF []	
DI [8]	*	OFF []	
DI [9]	*	OFF []	
DI [10]	*	OFF []	

[TYPE] CONFIG IN/OUT SIMULATE UNSIM

- 5 If the screen is displaying the Digital Outputs, press F3, IN/OUT to select the Digital Input Port.
- 6 Configure the I/O:
 - a Move the cursor to RACK, type the value, and press ENTER.
 - b Move the cursor to SLOT, type the value, and press ENTER.
 - c Move the cursor to START PT, type the value, and press ENTER.
- 7 **To determine if the assignment is valid**, press NEXT, >, and then press F2, VERIFY.
 - If the assignment is valid, the message, “Port assignment is valid,” is displayed.
 - If the assignment is not valid, the message, “Port assignment is invalid,” is displayed.

12.2.3 Sensor Setup

Table 12-1 lists and describes the items you must set in sensor setup. Use Procedure 12-5 to set up the sensor.

Table 12-1. Sensor Setup Screen

ITEM	DESCRIPTION
Data Port 1 Default: 1 Min: 1 Max: 999 Units: Integer	This item specifies the number of the port you have set up for I/O signal cable 1.
Data Port 2 Default: 2 Min: 1 Max: 999 Units: Integer	This item specifies the number of the port you have set up for I/O signal cable 2.
Motion Setup Menu	
Motion Speed Default: 200 mm/sec Min: 25 mm/sec Max: 500 mm/sec Units: Millimeters/Sec	This item defines the speed, in millimeters per second, of motions between two positions.
Pull Speed Default: 100 mm/sec Min: 25 mm/sec Max: 200 mm/sec Units: Millimeters/Sec	This variable specifies the maximum speed, in millimeters per second, that the robot will extend the string. It has a default value of 100 mm/sec.
Fine Speed Default: 2 mm/sec Min: 0.5 mm/sec Max: 5 mm/sec Units: Millimeters/sec	This specifies the robot pull speed, in millimeters per second, for sensor detection.
Pull Distance Default: 250 mm Min: 50 mm Max: 500 mm Units: Millimeters	This item specifies the maximum distance, in millimeters, that the robot will extend the string. Its default value is 250 millimeters.
Fitting Threshold Default: 2 mm Min: 0.1 mm Max: 5 mm Units: Millimeters	This item is used when computing the TCP value. Auto TCP does a sphere fitting for all recorded positions. If the sphere fitting error is greater than the fitting threshold, a message will be displayed to warn you of the difference. A large fitting error is usually caused by: 1. The string being prevented from moving during the robot motion, or 2. The robot mastering is incorrect. You should re-run Auto TCP after verifying that there are no obstacles within the Auto TCP workcell. If there is still an error, the robot mastering data has been changed.
Sensor Setup Menu	
Auto Radius Default: 250 mm Min: 25 mm Max: 1000 mm Units: Millimeters	This function is used to generate positions automatically. Refer to Section 12.4.2 for more details.

Table 12-1. (Cont'd) Sensor Setup Screen

ITEM	DESCRIPTION
TCP Error Threshold Default: 10 mm Min: 0 mm Max: 10 mm Units: Millimeters	This item allows you to specify a value that will be used to compare the old TCP to the new TCP. If the difference between the old TCP and the new TCP exceeds the value you have set, Auto TCP will set the DOUT port, and display a prompt box that will require a response from you. If the DOUT port is set to 0, there will be no Digital Output.
Error Report On Default: N/A Min: 1 Max: 999 Units: Integer	This specifies the number of the DOUT port the error occurred on.
Sensor Offset Default: 0.00 mm Min: 0.1 mm Max: 1000 mm Units: Millimeters	This item allows you to change the TCP value if the tool is offset from the sensor's TCP mounting fixture. These values should be set in relation to the robot's faceplate.

Procedure 12-5 Setting Up the Sensor

- Condition**
- The Auto TCP Hardware has been installed. (Procedure 12-1)
 - The String Sensor I/O cables have been installed. (Procedure 12-3)
 - The I/O Ports have been configured. (Procedure 12-4)
- Step**
- 1 Press **MENUS**.
 - 2 Select **UTILITIES**.
 - 3 Press **F1**, **[TYPE]**.
 - 4 Select **Auto TCP**. You will see a screen similar to the following.

```

ATCP Main Menu                                JOINT 10%

Current Tool: [1]
  X: -18.556 Y: 31.698 Z: 320.738
  W: -18.556 P: 0.000 R: 0.000
New Tool:   [1]
  X: -18.556 Y: 31.698 Z: 320.738
  W:  0.000 P:  0.000 R:  0.000
Sensor Setup:      INCOMPLETE
Orientation Setup  INCOMPLETE
Auto Setup:        INCOMPLETE
Record Data:       INCOMPLETE
[TYPE]  AUTO  EXEC  [DETAIL]  HELP

```


- 5 Move the cursor to Sensor Setup and press F4, [DETAIL] or ENTER. You will see a screen similar to the following.

ATCP Sensor Setup		JOINT 10%
		1/7
Data Port 1:	RDI[1]	
Data Port 2:	RDI[2]	
Motion Setup	COMPLETE	
Auto Radius:	200.000	
TCP Error Threshold:	.100	
Error Report On:	DOUT[1]	
Sensor Offset: [mm][World]		
X: 0.000	Y: 0.000	Z: 0.000
	[CHOICE]	HELP

- 6 Move the cursor to each item and set it as desired.
- 7 To perform motion setup,
- Move the cursor to Motion Setup.
 - Press ENTER. See the following screen for an example.

ATCP Sensor Setup		JOINT 10%
		1/5
Motion Speed:	200.000	
Pull Speed:	100.000	
Fine Speed:	2.000	
Pull Distance:	250.000	
Fitting Threshold::	2.0	
		HELP

- Move the cursor to each item and set it as desired.
 - When you are finished setting up motion items, press PREV.
- 8 When you are finished setting up the sensor, press PREV to display the Auto TCP menu.

Auto TCP Main Menu		JOINT 10%
MAIN		
Current Tool: [1]		
X: -18.556	Y: 31.698	Z: 320.738
W: 0.00	P: 0.000	R: 0.000
New Tool: [1]		
X: -18.556	Y: 31.698	Z: 320.738
W: 0.000	P: 0.000	R: 0.000
Sensor Setup:	COMPLETE	
Orientation Setup	INCOMPLETE	
Auto Setup:	INCOMPLETE	
Record Data:	INCOMPLETE	
[TYPE]	AUTO	EXEC UPDATE HELP

12.2.4 Testing Data Ports

After you have connected and configured the I/O signal cables, you should test the data ports to ensure they are functioning properly. Use Procedure 12-6 to test the Data Ports.

Procedure 12-6 Testing Data Ports

- Condition**
- The String Sensor is connected to the robot controller, and the attachment device has been connected to the tool. Refer to Procedure 12-1 .
 - You have installed the String Sensor I/O Cables. Refer to Procedure 12-3 .
 - You have configured the I/O Ports. Refer to Procedure 12-4 .
 - You have set up the String Sensor. Refer to Procedure 12-5 .
- Step**
- 1 Press MENUS.
 - 2 Select I/O.
 - 3 Press F1, [TYPE].
 - 4 Select the kind of I/O you are using for the data ports, either RDI or DI. See the following screen for an example of digital input.

I/O Digital In			JOINT	50 %
#	SIM	STATUS		1/256
DI [1]	*	OFF	[]
DI [2]	*	OFF	[]
DI [3]	*	OFF	[]
DI [4]	*	OFF	[]
DI [5]	*	OFF	[]
DI [6]	*	OFF	[]
DI [7]	*	OFF	[]
DI [8]	*	OFF	[]
DI [9]	*	OFF	[]
DI [10]	*	OFF	[]
[TYPE]	CONFIG	IN/OUT	SIMULATE	UNSIM

- 5 While slowly pulling the string on the string sensor, monitor the data port inputs. When both ports change to OFF, stop pulling the string.
- 6 While monitoring the Data Ports, slowly pull the string. Data port 1 should switch to ON before Data Port 2 does. If Data Port 2 switches to ON before Data Port 1, change the port assignments so that the first one that switched to ON is Data Port 1.

12.3 TCP ORIENTATION SETUP

You should perform TCP Orientation setup before Auto Setup. The orientation setup computes the TCP's orientation, or W, P, and R values. If the TCP orientation is not a critical factor in the work being performed by the robot, you do not have to perform TCP orientation setup, and can leave this entry as incomplete. The TCP's x, y, and z values will not be affected by TCP orientation setup.

The Orientation Setup menu allows you to set up the TCP's W, P and R values. Table 12-2 lists and describes the items in the Orientation Setup menu. Use Procedure 12-7 to set up TCP orientation.

Table 12-2. Orientation Setup Menu Items

ITEM	DESCRIPTION
Current Orientation	This item shows the actual orientation of the TCP, before Auto TCP has calculated a TCP.
New Orientation	This item shows how you want the TCP to be oriented after Auto TCP has calculated a TCP.
Reference Position	This item specifies the location within the UTOOL frame where the orientation of the TCP will originate.

Procedure 12-7 Setting Up TCP Orientation

- | | |
|---------------------------------------|---|
| Condition | <ul style="list-style-type: none"> ■ Auto TCP hardware is connected and properly configured. Refer to Section 12.2. ■ The Auto TCP String Sensor has been properly configured. Refer to Section 12.2. |
| To Record a Reference Position | <ol style="list-style-type: none"> 1 Press MENUS. 2 Select UTILITIES. 3 Press F1, [TYPE] 4 Select Auto TCP. You will see a screen similar to the following. |

```

Auto TCP Main Menu          JOINT 10%
                               1/7
Current Tool: [1]
  X: -18.556 Y: 31.698 Z: 320.738
  W: -18.556 P: 0.000 R: 0.000
New Tool:      [1]
  X: -18.556 Y: 31.698 Z: 320.738
  W:  0.000 P:  0.000 R:  0.000
Sensor Setup:          COMPLETE
Orientation Setup:    INCOMPLETE
Auto Setup:            INCOMPLETE
Record Data:          INCOMPLETE

[TYPE]  AUTO  EXEC  UPDATE  HELP

```

- 5 Move the cursor to Orientation Setup and press F4, DETAIL, or ENTER. You will see a screen similar to the following.

```

Auto TCP Orientation Menu      JOINT 10%
                                4/5
Current Orientation: [1]
W: -163.568 P: 9.932 R: -168.209

New Orientation:      [1]
W: 121.758 P: 70.871 R: -47.889

Reference Position:      RECORD
+Z: (Tool Frame): Front
+X: (Tool Frame): Up

                                UPDATE      [CHOICE]      HELP

```

- 6 Move the cursor to Reference Position.
- 7 Using the jog keys on the teach pendant, jog the robot to the desired reference point.
- 8 Press SHIFT, and F2, RECORD on the teach pendant to record the position.
- 9 To align the tool's z-axis to the world z-axis, move the cursor to +z.
- 10 Press F4, [CHOICE]. You will see a screen similar to the following.

Automatically Updating a UTOOL Frame

```

1 UP   (WORLD +Z)      5 LEFT (WORLD +Y)
2 DOWN (WORLD -Z)      6 RIGHT (WORLD -Y)
3 FRONT (WORLD +X)     7
4 BACK (WORLD -X)      8

Auto TCP Orientation Menu      JOINT 10%
                                4/5
W: -163.568 P: 9.932 R: -168.209

Reference Position:      RECORD
+Z: (Tool Frame): Front
+X: (Tool Frame): Up

                                UPDATE      [CHOICE]      HELP

```

- 11 Select a direction from the list shown appears at the top of the teach pendant screen. For example, to align the UTOOL frame's z axis to the World z Axis, move the cursor to UP (WORLD + Z).
- 12 Press F2, UPDATE.
- A message that asks if you want to change the UTOOL Data will be displayed.
- 13 Press F4, YES to update the UTOOL frame, or F5, NO to cancel the operation.

12.4 AUTO SETUP

The Auto Setup menu allows you to specify up to 24 points for Auto TCP to use in calculating the TCP. A point set is a set of four positions, where the second, third, and fourth positions are derived from the first position by their orientation angle. More points will allow a more accurate TCP to be calculated, but will require more time for the robot to move to all the points.

12.4.1 Tool Orientation

The Tool Orientation is used in the TCP calculation process, as a factor in computing a position in a point set. The second position of the point set will be calculated by rotating the first position of a point by 15 degrees on the faceplate axis.

The third position is calculated by rotating the first position of the point by 15 degrees on the y axis, and the fourth position is calculated by rotating the first position of the point by 15 degrees on the z axis.

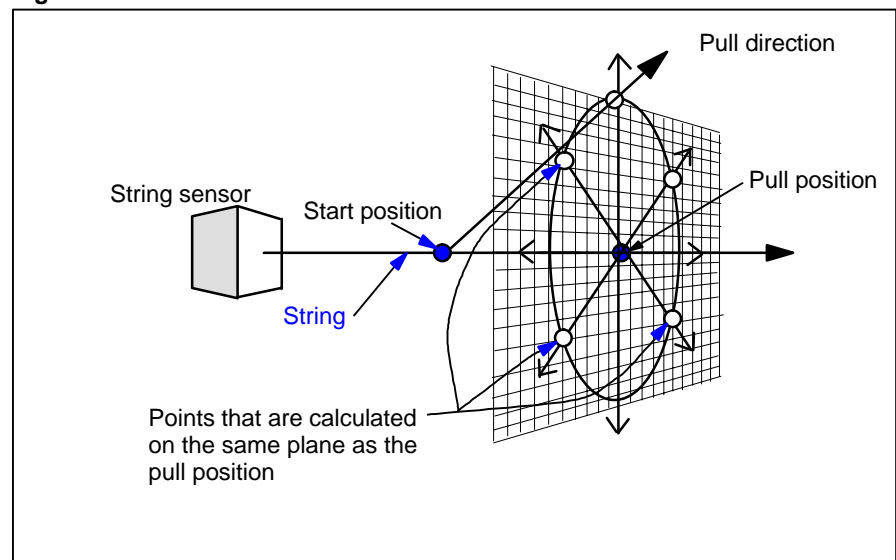
You have three options for the kind of tool orientation you want: Tight, Normal, and Large.

- The Tight orientation angle is 15 degrees.
- The Normal orientation uses a 30 degree angle of rotation for the point calculation.
- The large orientation uses a 45 degree angle to make the TCP calculation. A larger angle will allow a more accurate TCP to be calculated. You can specify an angle to use in the point calculation in the Position Set Menu.

12.4.2 Automatically Generating Positions

You must define the *start* and *pull* positions for Auto TCP to use in the calculation of a TCP. The start position is the point from which you want the robot to start pulling the string. The pull position is the point where you want the robot to pull the string to. The direction from the start position to the pull position is called the *pulling direction*, and all other points are generated on the same plane as the pull position. See Figure 12-4. It is recommended, for increased accuracy, that you record the start position close to the string sensor.

Figure 12-4. Auto TCP Point Generation



The Auto Radius option on the Sensor setup menu allows you to change how the points are calculated. A larger auto radius setting will require more space in the workcell for the robot to be moved within during the Auto TCP calculation process.

12.4.3 Manually Defining Positions

The Data Record menu allows you to define all the positions manually. The number of point sets you define will be used by Auto TCP in the calculation of the TCP.

12.4.4 Automatic and Manual Position Generation

Table 12-3 lists and describes the items you must set to generate Auto TCP points automatically. Use Procedure 12-8 to automatically or manually generate Auto TCP points.

12. AUTOMATIC TOOL CENTER POINT

Table 12-3. Auto Setup and Record Data Menu Items.

ITEM	DESCRIPTION
Auto Setup Menu	
Total Point Sets (3-6) Default: 6 Min: 3 Max: 6 Units: Integer	This item allows you to specify how many point sets Auto TCP will use to calculate the TCP. More point sets allow for a more accurate TCP, but the calculation will take longer to complete.
Tool Clearance Default: 15 Degrees Min: 15 Degrees Max: 45 Degrees Units: Degrees	This item specifies the Tool Orientation Angles: Tight, Normal, and Large
Start Position Default: Uninitialized Min: N/A Max: N/A Units: N/A	This item specifies the location where the robot begins pulling the Sensor string from during the calculation process. It is a position that you can jog the robot to, and record, or you can enter it manually.
Pull Position Default: Uninitialized Min: N/A Max: N/A Units:	This item specifies where the robot will pull the Sensor string to during the calculation process. It is a position that you can jog the robot to, and record, or you can enter it manually.
TPE Program Names Default: N/A Min: N/A Max: N/A Units: N/A	This item shows the programs that will automatically be executed by Auto TCP, if you choose to start Auto TCP by pressing the AUTO key.
Pre-AutoTCP TPE prog Default: Uninitialized Min: N/A Max: N/A Units: N/A	This program will automatically be executed by Auto TCP prior to TCP calculation, if you choose to start Auto TCP by pressing the AUTO key.
Post-AutoTCP TPE prog Default: Uninitialized Min: N/A Max: N/A Units: N/A	This program will automatically be executed by Auto TCP after TCP calculation, if you choose to start Auto TCP by pressing the AUTO key.
Record Data Menu	
Total Point Sets Default: 6 Min: 3 Max: 6 Units: Integer	This item allows you to specify how many point sets Auto TCP will use to calculate the TCP. More point sets allow for a more accurate TCP, but the calculation will take longer to complete.
Position Set Detail Screen	
Tool Rotation Default: 15 Degrees Min: -180 Degrees Max: 180 Degrees Units: Degrees	This item contains the Tool Orientation Angles: Tight, Normal, and Large. You can manually set the values.

Table 12-3. (Cont'd) Auto Setup and Record Data Menu Items.

ITEM	DESCRIPTION
Start Position Default: Uninitialized Min: N/A Max: N/A Units: N/A	This item specifies the location where the robot begins pulling the Sensor string from during the calculation process.
Position 1-4 Default: Uninitialized Min: N/A Max: N/A Units: N/A	This item specifies the Pull Position and points that will be calculated by Auto TCP. Position 1 is the Pull Position. The other three are points that will be calculated by applying tool rotation to position 1.

Procedure 12-8 Automatically Generating Positions

- Condition**
- Auto TCP hardware is connected and properly configured. (Section 12.2)
 - The Auto TCP String Sensor has been properly configured. (Section 12.2)
- Step**
- 1 Press MENUS.
 - 2 Select UTILITIES.
 - 3 Press F1, [TYPE].
- Perform Auto Setup**
- 4 Select Auto TCP. You will see a screen similar to the following.

```

Auto TCP Main Menu                                JOINT 10%
                                                    1/7
Current Tool: [1]
  X: -18.556 Y: 31.698 Z: 320.738
  W: -18.556 P: 0.000 R: 0.000
New Tool: [1]
  X: -18.556 Y: 31.698 Z: 320.738
  W: 0.000 P: 0.000 R: 0.000
Sensor Setup: Complete
Orientation Setup: Complete
Auto Setup: Incomplete
Record Data: Incomplete
[TYPE] AUTO EXEC UPDATE HELP
  
```


- 5 Move the cursor to Auto Setup and press ENTER. You will see a screen similar to the following.

Auto SETUP MENU		JOINT 10%	
			1/7
Total Point Sets (3-6):	4		
Tool Clearance:	NORMAL		
Start Position:	RECORD		
Pull Position:	RECORD		
TPE Program Names			
Pre-AutoTCP TPE prog:	PRE_PRGM		
Post-AutoTCP TPE prog:	POST_PRGM		
UPDATE	RECORD	MV_JNT	MV_LNR

1. Normal
2. Tight
3. Large

- 6 Move the cursor to Total Point Sets, type the number you want, and press the ENTER key.
- 7 Move the cursor to Tool Clearance, press F4, [CHOICE], and select the clearance you want.
- 8 Define the Start Position.
- a Move the cursor to Start Position.

NOTE By recording a Start Position closer to the sensor, you will increase the accuracy of the TCP and make it easier to re-calculate in the future.

- b Press and continue pressing the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
 - c Use the Jog keys to move the robots to the desired position.
 - d When the robot is at the Start position you want, press and hold SHIFT, and press F2, RECORD.
- 9 Define the Pull Position.
- a Move the cursor to Pull Position.

NOTE By recording a Pull Position closer to the sensor, you will increase the accuracy of the TCP and make it easier to recalculate it in the future. The distance between the Start Position and Pull Position should be at least 10 millimeters.

- b Press and continue pressing the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
- c Use the jog keys to move the robot to the desired position.
- d When the robot is at the Start position you want, press and hold SHIFT, and press F2, RECORD.

- 10 Press F2, [UPDATE], to generate all the positions automatically. If this step is not done before running Auto TCP, execution of the Auto TCP program will fail.
- 11 Define the Teach Pendant Program names:
 - a Move the cursor to Pre-AutoTCP.
 - b Press F4, [CHOICE].
 - c Select the program you want to use.
 - d Move the cursor to Post-AutoTCP, and repeat steps 11b and 11c.
- 12 When you are finished, press PREV to return to the Auto TCP menu. You will see a screen similar to the following.

```

Auto TCP Main Menu                               JOINT 10%
                                                1/7
Current Tool: [1]
X:  -18.556 Y:  31.698 Z: 320.738
W:   0.000 P:   0.000 R:  0.000
New Tool:    [1]
X:  -18.556 Y:  31.698 Z: 320.738
W:   0.000 P:   0.000 R:  0.000
Sensor Setup:          COMPLETE
Orientation Setup:    COMPLETE
Auto Setup:            COMPLETE
Record Data:          INCOMPLETE

[TYPE]  AUTO   EXEC  UPDATE   HELP

```

- Record Data**
- 13 Move the cursor to Record Data and press ENTER or F2, DETAIL. You will see a screen similar to the following.

```

GUNOFF                               LINE 0
MAIN
Auto TCP DATA                         JOINT 10%
                                                1/8

Total Point Sets:                       4

Position set 1:                          DONE
Position set 2:                          DONE
Position set 3:                          DONE
Position set 4:                          DONE
Position set 5:                          UNINIT
Position set 6:                          UNINIT

AUTO                                     HELP

```

- 14 Move the cursor to Total Point Sets and press F2, AUTO. Auto TCP will automatically generate all the point sets.

NOTE You can also automatically generate positions if the Tool Rotation, Number of Positions, Start Position, and Position 1 have all been set in Position set 1.

Manually Recording Data

- 15** Move the cursor to Position Set and press ENTER, or F4, DETAIL to set Position Set 1. You will see a screen similar to the following:

GUNOFF	LINE 0
MAIN	
Auto TCP POINT	JOINT 10%
	1/8
Number of Positions:	4
Tool Rotation:	SET
Start Position:	RECORD
Position NO 1:	RECORD
Position NO 2:	RECORD
Position NO 3:	RECORD
Position NO 4:	RECORD
AUTO	HELP

- 16** Move the cursor to Start Position.
- 17** Jog the robot to the desired position.
- 18** Press and hold SHIFT and press F2, RECORD to record the position. Repeat Steps 16 and 17 for each of the Positions.

NOTE You can manually define the Start position, and position 1, and have position 2, 3 and 4 automatically generated. To do this, perform steps 19 and 20. The Tool Orientation must be set for position 2, 3, and 4 to be automatically generated.

- 19** Move the cursor to Number of Positions.
- 20** Press AUTO. Positions 2, 3 and 4 will be automatically generated.
- 21** To verify automatically generated positions, you can move the cursor to each of the four positions, and use MV_JNT or MV_LNR to verify Positions 1 through 4.

12.5 AUTOMATICALLY CALCULATING A NEW UTOOL TCP

After you have installed Auto TCP hardware and sensors, and performed all necessary configuration and setup procedures, you can use Auto TCP to calculate a new UTOOL TCP automatically.

There are two ways to run Auto TCP programs:

- AUTO key execution
- EXEC key execution

You can also execute Auto TCP through the PLC by setting up the style menu and then calling the AUTOTCP.TP program.

AUTO Execution

If you execute Auto TCP by pressing the AUTO key, the Pre-AutoTCP program is executed before a New Tool TCP is calculated, and the Post-AutoTCP program is executed afterwards. A warning will also prompt you if the New Tool TCP is considerably different from the previous one. When you run Auto TCP using the AUTO Execution method, exiting the Auto TCP menu will not stop Auto TCP's execution.

EXEC Execution

If you execute Auto TCP by pressing the EXEC key, the Pre-AutoTCP and Post-AutoTCP programs will not be executed. Also, you will not be warned if a radically different New Tool TCP has been calculated. When you run Auto TCP using the EXEC Execution method, exiting the Auto TCP to other menus will stop Auto TCP's execution.

Changing New UTOOL Values

You can change the x, y, z, w, p, and r values of the new tool. The values that you enter will be used by Auto TCP as the initial values for the next Auto TCP New Tool TCP calculation.

If you set all the values to zero, Auto TCP will move the robot throughout the calculation sequence twice. The first set of motions will be performed at small rotational angles to get an estimated TCP. Auto TCP will then repeat the motions using the rotational angles specified in the Auto Setup menu to get an actual TCP. Auto TCP functions in this way in order to prevent damage to equipment or injury to personnel.

Table 12-4 lists and describes the items you will use on the Auto TCP menu during automatic TCP calculation. Use Procedure 12-9 to calculate a new UTOOL TCP automatically.

Table 12-4. Auto TCP Menu Items

ITEM	DESCRIPTION
Auto TCP Menu	
Current Tool Default: 1 Min: 1 Max: 6 Units: Integer	The Current Tool will show the \$MNUTOOL[n] current value, where 'n' is a value between 1 and 6. You can change the Current Tool number to calculate any of the six available \$MNUTOOL frames.
New Tool Default: N/A Min: N/A Max: N/A Units: N/A	The New Tool shows the coordinates for the new TCP that have been calculated by Auto TCP.

Procedure 12-9 Automatically Calculating a New UTOOL TCP

- 1 Press MENUS.
- 2 Select UTILITIES.
- 3 Press F1, [TYPE].
- 4 Select Auto TCP. You will see a screen similar to the following.

```

Auto TCP Main Menu                                JOINT 10%
                                                1/7
Current Tool: [1]
  X: -18.556 Y: 31.698 Z: 320.738
  W: -18.556 P: 0.000 R: 0.000
New Tool:    [1]
  X: -18.556 Y: 31.698 Z: 320.738
  W:  0.000 P:  0.000 R:  0.000

Sensor Setup:          COMPLETE
Orientation Setup:    COMPLETE
Auto Setup:            COMPLETE
Record Data:          COMPLETE

[TYPE]  AUTO  EXEC  UPDATE  HELP

```

**WARNING**

Be sure to step through all programs that use automatically generated tool center points at a slow speed before you run them at production speed. Otherwise, you could injure personnel or damage equipment.

- 5 Run the AutoTCP programs.
 - To have Auto TCP run the Pre-AutoTCP and Post-AutoTCP programs, go to Step 6.
 - To run Auto TCP without the Pre-AutoTCP and Post-Auto TCP programs, go to Step 7.

NOTE Auto TCP will generate only the x, y, and z values of the new tool TCP. The orientation is not computed. Use the Orientation Setup menu to set up TCP orientation. Refer to Section 12.3

-
- AUTO TCP Generation**
- 6** To calculate a TCP using the AUTO function key,
- a** Press F2, AUTO.
The UTOOL will be calculated. When the UTOOL has been successfully computed, the message “Utool successfully computed” will be displayed, along with any errors that might have occurred during the calculation process.
 - b** **If there are sphere fitting errors that exceed the Fitting Error Threshold defined in the Motion submenu**, check that the string has not been prevented from moving during the robot motion. If the string sensor is moving properly, then the robot mastering is incorrect. You should rerun Auto TCP after verifying that there are no obstacles within the Auto TCP workcell. If there is still an error, the robot mastering data has been changed. Refer to Appendix E for information on mastering the robot.
- EXEC TCP Generation**
- 7** To calculate the TCP using the EXEC function key,
- a** Press the EXEC key on the teach pendant.
The UTOOL will be calculated. When the UTOOL has been successfully computed, the message “Utool successfully computed” will be displayed, along with any errors that might have occurred during the calculation process.
 - b** **If there are sphere fitting errors that exceed the Fitting Error Threshold defined under the Motion submenu**, check that the string has not been prevented from moving during the robot motion. If the string sensor is moving properly, then the robot mastering is incorrect. You should rerun Auto TCP after verifying that there are no obstacles within the Auto TCP work cell. If there is still an error, the robot mastering data has been changed.
- Change Values of New UTOOL**
- 8** If you want to change the x, y, z, w, p, and r values of the new UTOOL that was calculated by Auto TCP, do the following:
- a** Move the cursor to the positional component you want to change (x, y, z, w, p, r).
 - b** Type the value you want and press ENTER.
 - c** Repeat Steps 8a and 8b for each positional component you want to change.
The values that you enter will be used by Auto TCP as the initial values for the next Auto TCP New Tool TCP calculation. If you set all the values to zero, Auto TCP will move the robot throughout the calculation sequence twice. The first set of motions will be performed at small rotational angles to get an estimated TCP. Auto TCP will then repeat the motions using the rotational angles specified in the Auto Setup screen to get an actual TCP.
- 9** After a TCP value has been calculated by Auto TCP, press F4, UPDATE to copy the new TCP value to MNUTOOL.
You will see a message asking you if you want to update MNUTOOL with the new TCP. If you select F4, YES, Auto TCP will put the previous value into the alarm log, so that you have a backup copy of the previous TCP value.

A

ERROR CODES AND RECOVERY

Topics In This Appendix		Page
Overview	This section contains information on the ALARMS screen and the items that make up an error code	A-3
	• Facility Name and Code	A-6
	• Severity Descriptions	A-7
	• Error Message Text	A-9
General Error Recovery Procedures	This section contains procedures for recovering from certain errors	A-11
	• Overtravel Release	A-11
	• Hand Breakage Recovery	A-13
	• Pulse Coder Alarm Recovery	A-14
Error Codes	This section contains error codes, possible causes, and remedies, listed in alphabetical order.	A-16

Errors occur because of

- Hardware problems – a broken cable or tooling
- Software problems – incorrect program or data
- External problems – an open safety door or an overtravel has occurred

Depending on the severity of the error, you must take certain steps to recover from it.

Use Procedure A-1 as the recommended error recovery procedure.

Procedure A-1 Error Recovery Recommendation

Condition ■ An error has occurred.

- Step**
- 1 Determine the cause of the error.
 - 2 Correct the problem which caused the error.
 - 3 Release the error.
 - 4 Restart the program or robot.

If the basic recovery procedures do not clear the error, try restarting the controller. Refer to Table A-1 for the methods of starting the controller. First try a cold start. If cold start does not solve the problem, try a controlled start. If the problem still exists, refer to the appropriate application-specific *FANUC Robotics SYSTEM R-J2 Software Installation Manual* to reload software if necessary.

Table A-1. Start Methods

Start Method	Description	Procedure
Cold start (START COLD)	Initializes changes to system variables Initializes changes to I/O setup Displays the UTILITIES Hints screen Recovers the C-WORK temporary memory area	On the teach pendant, press and hold the PREV and NEXT keys and press the ON button on the operator box. After the BMON> prompt appears on the teach pendant screen, release the keys. Press F1, COLD, and press ENTER. Press F5, START, and press ENTER.
Controlled start (START CTRL)	Allows you to set up application specific information Allows you to install options and updates Allows you to save specific information Allows you to start KCL Allows you to print teach pendant screens and the current robot configuration Allows you to unsimulate all I/O Does not allow you to load teach pendant programs	On the teach pendant, press and hold the PREV and NEXT keys and press the ON button on the operator box. After the BMON> prompt appears on the teach pendant screen, release the keys. Press F2, CTRL, and press ENTER. Press F5, START, and press ENTER.
Controlled 2 start (START CTRL2)	Updates memory Allows you to load teach pendant programs	On the teach pendant, perform a controlled start. Select 4, EXIT, and press ENTER. Press F4, YES. Press FCTN. Select START (CTRL2). When the CTRL2 start has completed, press FCTN and select START (COLD).

A.1 OVERVIEW

An error message consists of:

- The facility name and error code number
- The severity of the error
- The message text of the error

The error message will be displayed as follows:

FACILITY_NAME - ERROR_CODE_NUMBER Error message text

The Alarm Log screen displays a list of errors that have occurred. There are two ways to display alarms:

- **Automatically** using the Active Alarm screen. This screen displays only active errors (with a severity other than WARN) that have occurred since the last controlled reset
- **Manually** using the History Alarm screen. This screen displays up to the last 100 alarms, regardless of their severity. You can also display detailed information about a specific alarm.

Use Procedure A-2 to display the Alarm Log screen.

Procedure A-2 Displaying the Alarm Log Screen

Condition

- To display the Active Alarm screen automatically,
 - The system variable \$ER_AUTO_ENB must be set to TRUE. Then you must have performed a cold start.
 - An error, whose severity is either PAUSE or ABORT must have occurred.

Automatic Display of Active Alarm Screen

The following screen will automatically be displayed. It lists all errors with a severity other than WARN, that have occurred since the last controller RESET. The most recent error is number 1.

```

SRVO-007 External emergency stop
TEST1 LINE 15 ABORTED
Alarm: ACTIVE WORLD 100 %
1/100
1 SRVO-007 External emergency stop

[ TYPE ] HIST
  
```

- 1 To toggle between the Active Alarm screen and Hist Alarm screen, press F3 (ACTIVE or HIST).
- 2 To disable the automatic display of all errors with a certain severity type, modify the value of the system variable \$ER_SEV_NOAUTO[1-5]. Then these errors will still be logged in the Active Alarm screen, but they will no longer force the screen to immediately become visible. Refer to Section A.1.2 for more information.

- 3 To disable the automatic display of a specific error code, modify the \$ER_NOAUTO.\$noalm_num and \$ER_NOAUTO.\$ser_code system variables. These errors will still be logged in the Active Alarm screen, but they will no longer force the screen to immediately become visible. Refer to the *SYSTEM R-J2 Software Reference Manual* for more detailed information about how to set these variables.
- 4 To display the screen that occurred immediately before the alarm, press RESET. If you have toggled between HIST and ACTIVE, the previous screen might not be available.

When there are no active alarms (the system is not in error status), the following message will be displayed on the Active Alarm screen.

There are no active alarms.

Press F3(HIST) to enter alarm history screen.

NOTE When you reset the system by pressing the RESET key, the alarms displayed on this screen are cleared.

Manual Display of History Alarm Screen

- 5 Press F3, HIST.
- 6 Press F1, [TYPE].
- 7 Select Alarm Log. The alarm log will be displayed. This lists all errors. See the following screen for an example.

```

SRVO-007 External emergency stop
TEST1          LINE 15          ABORTED
Alarm: HIST          WORLD 100 %
                                1/100
 1 SRVO-007 External emergency stop
 2 SRVO-001 Operator panel emergency st
 3 R E S E T
 4 SRVO-029 Robot calibrated (Group:1)
 5 SRVO-001 Operator panel emergency st
 6 SRVO-012 Power fail recovery
 7 INTP-127 Power fail detected
 8 SRVO-047 LVAL alarm (Group:1 Axis:5)
 9 SRVO-047 LVAL alarm (Group:1 Axis:4)
10 SRVO-002 Teach pendant emergency stop

[ TYPE ]          ACTIVE CLEAR HELP

```

NOTE The most recent error is number 1.

- To display the complete error message that does not fit on the screen, press F5, HELP, and the right arrow key on the teach pendant.
- To display the cause code for an error message, press F5, HELP. Cause codes provide further information about the cause of the error. If the specified error has a cause code, the cause code message is displayed immediately below the error line, on the status line. When you press RESET, the error and cause code disappears and the status line is redisplayed.

- 8 To display the motion log**, which lists only motion-related errors, press F1, [TYPE], and select Motion Log.
- 9 To display the system log**, which displays only system errors, press F1, [TYPE], and select System Log.
- 10 To display the application log**, which displays only HandlingTool errors, press F1, [TYPE], and select Appl Log.
- 11 To display more information about an error**, move the cursor to the error and press F5, HELP. The error help screen displays information specific to the error you selected, including the severity. If the error has a cause code, the cause code message will be displayed. When you are finished viewing the information, press PREV.
- 12 To remove all of the error messages displayed on the screen**, press and hold SHIFT and press F4, CLEAR.

A.1.1 Facility Name and Code

The facility name and code identify the type of error that occurred. Facility information is displayed at the beginning of the error code. For example:

PROG-048 Shift released while running

In the above example, the facility name PROG corresponds to facility code 3. The error code number is 048. Facility codes are used in error handling from a KAREL program. The facility codes are listed in Table A-2.

Table A-2. Error Facility Codes

Facility Name	Facility Code (Decimal)	Facility Code (Hexadecimal)	Description
APSH	38	0x26	Application shell
CD	82	0x52	Coordinated motion
CMCC	55	0x37	IntelliTrak
CMND	42	0x2a	Command processor
COND	4	0x4	Condition handler
DICT	33	0x21	Dictionary processor
DNET	76	0x4c	DeviceNet
ELOG	5	0x5	Error logger
FILE	2	0x2	File system
FLPY	10	0xa	Serial floppy disk system
FRSY	85	0x55	FROM device system
HOST	67	0x43	Host communications
HRTL	66	0x42	Communication tag
INTP	12	0xc	Interpreter errors
JOG	19	0x13	Manual jog task
LANG	21	0x15	Language utility
LNTK	44	0x2c	Line tracking
MACR	57	0x39	MACRO option
MCTL	6	0x6	Motion control manager
MEMO	7	0x7	Memory manager
MOTN	15	0xf	Motion subsystem
OPTN	65	0x41	Option installation
PALT	26	0x1a	Palletizing
PRIO	13	0xd	Digital I/O subsystem
PROG	3	0x3	Interpreter
PWD	31	0x1f	Password
QMGR	61	0x3d	Queue
ROUT	17	0x11	Interpreter built-ins
SCIO	25	0x19	Syntax checking for teach pendant programs
SENS	58	0x3a	Sensor
SRVO	11	0xb	FLTR&SERVO in motion sub-system
SSPC	69	0x45	Space check
SYST	24	0x18	Facility code of system
TPIF	9	0x9	Teach pendant user interface
VAR5	16	0x10	Variable manager subsystem
WNDW	18	0x12	Window I/O manager

A.1.2 Severity Descriptions

The severity of the error indicates how serious the error is.

NOTE You can displayed the severity of the error code on the ALARM screen. Refer to Procedure A-2 .

\$ER_SEV_NOAUTO[1-5] System Variable

The \$ER_SEV_NOAUTO[1-5] system variable enables or disables the automatic display of all error codes with a particular *severity*. This is used in conjunction with the \$ER_AUTO_ENB system variable.

SEVERITY	\$ER_SEV_NOAUTO[1-5]
PAUSE	[1]
STOP	[2]
SERVO	[3]
ABORT	[4]
SYSTEM	[5]

WARN

WARN errors only warn of potential problems or unexpected circumstances. They do not directly affect any operations that might be in progress. If a WARN error occurs, you should determine what caused the error and what, if any, actions should be taken.

PAUSE

PAUSE errors pause program execution but allow the robot to complete its current motion segment, if any are in progress. This error typically indicates that some action must be taken before program execution can be resumed. PAUSE errors cause the operator panel FAULT light to go on and the teach pendant FAULT LED to go on.

Depending on the action that is required, you might be able to resume a paused program at the point where the PAUSE error occurred after you have corrected the error condition. If the program can be resumed, you can either select the RESUME function key or press the operator CYCLE START button, depending on the position of the REMOTE keyswitch.

STOP

STOP errors pause program execution and stop robot motion. When a motion is stopped, the robot decelerates to a stop and any remaining part of the current motion segment is saved, meaning the motion can be resumed. STOP errors usually indicate that some action must be taken before the motion and program execution can be resumed.

Depending on the action that is required, you might be able to resume the motion and program execution after correcting the error condition. If the motion and program can be resumed, you can either select the RESUME function key or press the operator CYCLE START button depending on the position of the keyswitch.

SERVO

SERVO errors shut off the drive power to the servo system and pause program execution. SERVO errors cause the operator panel FAULT light to go on and the teach pendant FAULT LED to go on.

SERVO errors are usually caused by hardware problems and could require trained service personnel. However, some SERVO errors require you to reset the servo system by pressing the operator panel FAULT RESET button or the teach pendant RESET key. Others require a cold start of the controller.

ABORT

ABORT errors abort program execution and STOP robot motion. When an ABORT error occurs, the robot decelerates to a STOP and the remainder of the motion is canceled. An ABORT error indicates that the program has a problem that is severe enough to prevent it from continuing to run.

You will need to correct the problem and then restart the program. Depending on the error, correcting the problem might mean editing the program or modifying the data.

SYSTEM

SYSTEM errors usually indicate a system problem exists that is severe enough to prevent any further operation. The problem could be hardware or software related.

You will need the assistance of trained service personnel to correct SYSTEM errors. After the error has been corrected, you will need to reset the system by turning off the robot, waiting a few seconds, and turning on the robot.

If a program was executing when the error occurred, you will need to restart the program.

NONE

NONE errors can be returned as status from some KAREL built-in routines and can also be used to trigger KAREL condition handlers. NONE errors are not displayed on the teach pendant or CRT/KB. They also are not displayed on the alarm log screen. NONE errors do not have any effect on programs, robot motion, or servo motors.

Table A-3 summarizes the effects of error severities.

Table A-3. Effects of Error Severity

Severity	Program	Robot Motion	Servo Motors
WARN	No effect	No effect	No effect
PAUSE	Paused	The current move is completed then the robot stops.	No effect
STOP	Paused	Decelerated STOP, motion retained	No effect
SERVO	Paused	Decelerated STOP, motion retained	Power shutdown
ABORT	Aborted	Emergency STOP, motion canceled	No effect
SYSTEM	Aborted	Emergency STOP, motion canceled	Power shut down Requires turning off/turning on the robot
NONE	No effect	No effect	No effect

A.1.3 Error Message Text

The message text describes the error that has occurred. Message text is displayed at the end of the error code. For example:

```
PROG-048 Shift released while running
```

Some error messages might contain cause codes, percent (%) notation, or hexadecimal notation. For more information on displaying cause codes, refer to Procedure A-2 .

Percent Notation (%)

A percent sign followed by the letter *s* (%s) indicates that a string, representing a program name, file name, or variable name, actually appears in the error message when the error occurs.

A percent sign followed by the letter *d* (%d) indicates that an integer, representing a program line number or other numeric value, actually appears in the error message when the error occurs.

For example:

```
INTP-327 (%^s, %d^5) Open file failed
```

When this error occurs, the actual name of the file that could not be opened will appear on the teach pendant error line instead of %s. The actual program line number on which that error occurred will appear on the teach pendant error line instead of %d.

Hexadecimal Notation

Hexadecimal notation is used to indicate the specific axes in error, when one or more axes are in error at the same time.

Most robots have interaction limits, in addition to normal joint limits. Even when all axes are within their respective limits an error might occur. This could possibly be caused by the interaction between multiple axes. In this case, hexadecimal notation can help you to find the specific axis in error. For example:

```
MOTN-017 limit error (G:1 A:6 Hex)
```

The number after the “**A**” is the hexadecimal digit that shows which axes are out of limit. The “**Hex**” indicates that the axis numbers are in hexadecimal format. Figure A-1 lists the sixteen hexadecimal digits and the corresponding axes that are in error.

NOTE Hexadecimal digits for the decimal values of 10 through 15 are represented by the letters A through F respectively. Refer to Figure A-1.

To determine which axes are in error, you must evaluate each digit in the error message separately. Refer to Figure A-1.

NOTE If only one number appears in the error message after the “A:”, you must read it as the first digit.

Figure A-1. Hexadecimal Error Message Display

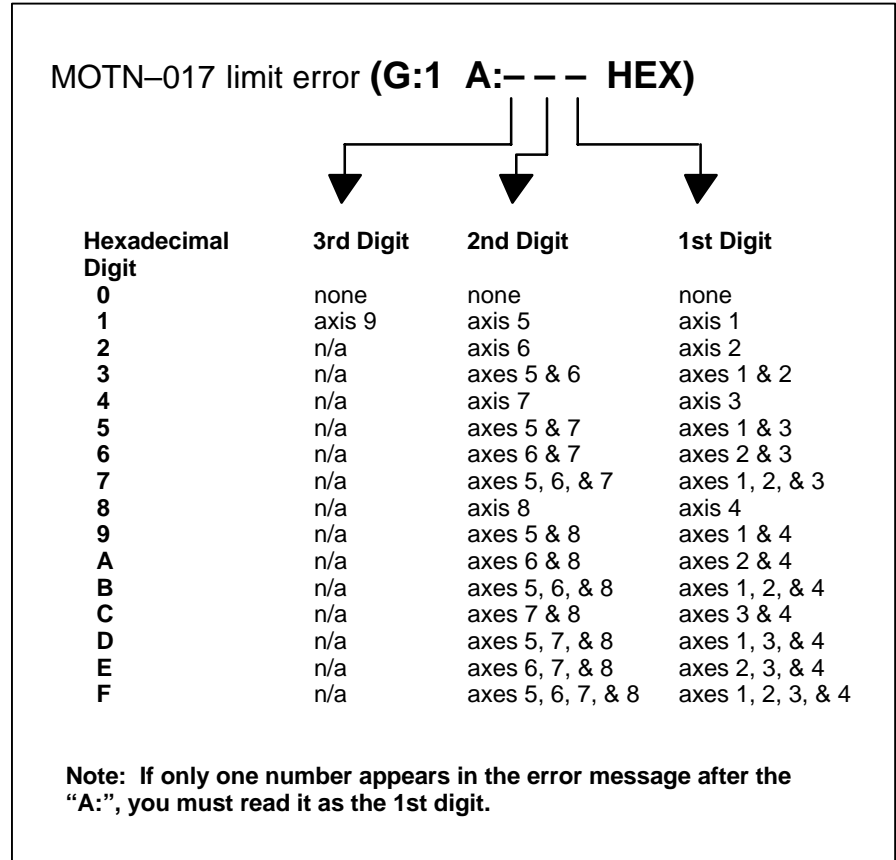


Table A-4 contains some examples of how to interpret Hexadecimal notation in an error message.

Table A-4. Hexadecimal Notation and Axis in Error Examples

Error	Explanation
MOTN-017 (G:1 A:6 Hex)	Axes 2 and 3 are out of their interaction limit.
MJOG-013 (G:1 A:20 Hex)	Axis 6 jogged to limit.
MOTN-017 (G:1 A:100 Hex)	Axis 9 limit error.

A.2 GENERAL ERROR RECOVERY PROCEDURES

This section contains procedures for recovery from certain errors. These errors are:

- Overtravel release
- Hand breakage recovery
- Pulse coder alarm

A.2.1 Overtravel Release

An overtravel error occurs when one or more of the robot axes moves beyond the software motion limits. When this happens one of the overtravel limit switches is tripped and the system does the following:

- Shuts off drive power to the servo system and applies robot brakes
- Displays an overtravel alarm error message
- Lights the operator panel FAULT light
- Turns on the teach pendant FAULT status indicator
- Limits motion for the axes involved in the overtravel

Use Procedure A-3 to recover from an overtravel error.

Procedure A-3 Recovering from an Overtravel Error

- Condition** ■ An axis (or axes) are in overtravel and the overtravel alarm has occurred. If you are jogging in JOINT the axis number indicating the axis (or axes) in an overtravel will be displayed in the error log.
- Step**
- 1 Press MENU.
 - 2 Select MANUAL FCTNS.
 - 3 Press F1, [TYPE].
 - 4 Select OT_RELEASE. You will see a screen similar to the following. The axis that is overtraveled will display TRUE in either OT_MINUS or OT_PLUS.

MANUAL OT Release E1 JOINT 10 %		
AXIS	OT MINUS	OT PLUS
1	FALSE	TRUE
2	FALSE	FALSE
3	FALSE	FALSE
4	FALSE	FALSE
5	FALSE	FALSE
6	FALSE	FALSE
7	FALSE	FALSE
8	FALSE	FALSE
9	FALSE	FALSE

[TYPE] RELEASE

TRUE indicates an axis is in overtravel. **FALSE** indicates an axis is not in overtravel.

If the robot is calibrated

- 5 Move the cursor to the OT PLUS or OT MINUS value of the axis in overtravel.
- 6 Press F2, RELEASE. The value of the overtraveled axis should change back to FALSE.
- 7 **If the robot is calibrated**, you will see the message “Can’t Release OT. Press HELP for detail.”
 - a If you press F5, HELP, you will see a screen similar to the following.

```

MANUAL OT Release      E1      JOINT 10 %
When robot is calibrated, overtravel
cannot be released. Press SHIFT &
RESET to clear the error, and jog out
of the overtravel condition.

[ TYPE ] RELEASE

```

NOTE For the following steps, press and **hold down the SHIFT key** until you have completed Steps b through d.

- b Press and continue pressing SHIFT and press F2, RESET. Wait for servo power.
- c Continuously press and hold the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
- d Jog the overtraveled axis off the overtravel switch. When you have finished jogging, **you can release the SHIFT key**.

NOTE If you accidentally release the shift key during Steps b through d, you will have to repeat them.

If the robot is not calibrated

- 8 **If the robot is not calibrated**, perform the following steps:

NOTE For the following steps, press and **hold down the SHIFT key** until you have completed Steps a through d.

- a Press and continue pressing SHIFT and press F2, RESET. Wait for servo power.
- b Press COORD until you select the JOINT coordinate system.
- c Continuously press and hold the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
- d Jog the overtraveled axis off the overtravel switch. When you have finished jogging, **you can release the SHIFT key**.

NOTE If you accidentally release the shift key during Step 8, you will need to repeat Step 8.

- 9 Turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.
- 10 Check CRM11 connection on axis control PCB if the robot is not in an actual overtravel condition.

A.2.2 Hand Breakage Recovery

A hand breakage error occurs when the hand breakage detection switch is tripped on robots equipped with hand breakage hardware. The switch is tripped when the robot tool strikes an obstacle, which could possibly cause the tool to break. The system

- Shuts off drive power to the servo system and applies robot brakes
- Displays an error message indicating that the hand is broken
- Lights the operator panel FAULT light
- Lights the teach pendant FAULT LED

The status of the hand breakage detection switch is displayed on the STATUS Safety Signals screen.

Use Procedure A-4 to recover from a hand breakage.

Procedure A-4 Recovering from a Hand Breakage

Condition	■ The hand breakage error message is displayed.
Step	<ol style="list-style-type: none"> 1 If you have not already done so, continuously press and hold the DEADMAN switch and turn the teach pendant ON/OFF switch to ON. 2 Hold down the SHIFT key and press RESET. The robot can now be moved. 3 Jog the robot to a safe position. 4 Press the EMERGENCY STOP button. 5 Request a trained service person to inspect and, if necessary, repair the tool. 6 Determine what caused the tool to strike an object, causing the hand to break. 7 If the hand breakage occurred while a program was being executed, you might need to reteach positions, edit the program, or move the object that was struck. 8 Test run the program if it has been edited, if new positions have been recorded, or if objects in the work envelope have been moved.

A.2.3 Pulse Coder Alarm Recovery

If the pulse counts at power up do not match the pulse counts at power down, a pulse mismatch error occurs for each motion group and each axis. Use Procedure A-5 to reset a pulse coder alarm.

Procedure A-5 Using the Mastering Routine

- Step**
- 1 Press SYSTEM.
 - 2 Press F1, [TYPE].
 - 3 Select Master/Cal.

If Master/Cal is not listed on the [TYPE] menu, do the following; otherwise, continue to Step 4.

 - a Select VARIABLE from the [TYPE] menu.
 - b Move the cursor to \$MASTER_ENB.
 - c Press the numeric key “1” and then press ENTER on the teach pendant.
 - d Press F1, [TYPE].
 - e Select Master/Cal. You will see a screen similar to the following.

```

SYSTEM Master/Cal                                JOINT 10%
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE

Press 'ENTER' or number key to select.

[ TYPE ]   LOAD   RES_PCA                        DONE
  
```

- 4 Press F3, RES_PCA. You will see a screen similar to the following.

```
SYSTEM Master/Cal JOINT 10%
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.

Reset pulse coder alarm? [NO]
[ TYPE ] YES NO
```

- 5 Press F4, YES. You will see a screen similar to the following.

```
SYSTEM Master/Cal JOINT 10%
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Pulse coder alarm reset!

[ TYPE ] LOAD RES_PCA DONE
```

A.3 ERROR CODES

Error codes in this section are listed alphabetically. Percent signs (%) followed by a letter or letters indicate that a program name, file name, or variable name actually appears in the error message when the error occurs.

For example, the error message for APSH-015 is listed in this section as **Program %s not loaded.** When this error occurs, the actual name of the program that was not loaded will appear on the teach pendant error line instead of %s.

If you receive an error that does not appear in this section, write down all of the events that lead to the error. Try to cold start the controller. If the error still occurs, follow your normal procedure for unresolved errors.

APSH Error Codes (ID = 38)

APSH-000 WARN Unknown error (APSH0)

Cause: An internal error has occurred.

Remedy: Cold start the controller

APSH-001 PAUSE %s

APSH-002 WARN REMOTE switch must be on LOCAL

Cause: The REMOTE keyswitch is incorrectly set for the the source of the run command

Remedy: Switch REMOTE keyswitch as directed on error message

APSH-003 WARN REMOTE switch must be REMOTE

Cause: The REMOTE keyswitch is incorrectly set for the the source of the run command

Remedy: Switch REMOTE keyswitch as directed on error message

APSH-004 WARN FAULT must be reset

Cause: There is an active fault, which prevents the run request from executing

Remedy: Clear the source of the fault, and press FAULT RESET.

APSH-005 WARN Step mode must be disabled

Cause: Step mode is enabled, which prevents the run request from executing

Remedy: Disable step mode by pressing the STEP hardkey.

APSH-006 WARN UOP is not the master device

Cause: The UOP is not the master device, which prevents the run request from executing

Remedy: Set \$RMT_MASTER=5

APSH-007 WARN WARNING - System is in dry run

Cause: This is just a warning to notify the user that a production job is running with the process disabled.

Remedy: None needed.

APSH-008 WARN WARNING - Machine lock is on

Cause: This is just a warning to notify the user that a production job is running with machine lock on.

Remedy: None needed.

APSH-010 WARN Job queue is full

Cause: The job queue cannot accept the next job because it is full.

Remedy: Manually edit the job queue to delete any unneeded jobs.

APSH-011 WARN No jobs in queue

Cause: A request to run the next job in the queue came in, but the queue is empty.

Remedy: Check the external device to make sure that a job number was correctly sent to the controller, or manually edit the job queue to insert a job.

APSH-012 WARN Robot motion is not enabled

Cause: Robot motion and/or program execution cannot occur unless UOP input 8 is

Remedy: Check UOP I/O setup. UOP input 8 must be high.

APSH-013 WARN Safety fence is open

Cause: The safety fence is open, as indicated by UOP input [3] being low.

Remedy: Close the fence, and/or check UOP I/O setup. UOP input [3] is high when fence is closed.

APSH-014 WARN Teach Pendant is enabled

Cause: The teach pendant is enabled which prevents the run request from being executed.

Remedy: Disable the teach pendant.

APSH-015 WARN Program %s not loaded

Cause: The program name that the shell was requested to run is not loaded on the controller.

Remedy: Load or create the program. Check \$SHELL_CFG variables to make sure job name variables are correct.

APSH-016 WARN WARNING - Running at < 100%%

Cause: This is just a warning to notify the user that a production job is running at less than 100% override.

Remedy: None needed.

APSH-017 WARN Program already running

Cause: The shell detected a start signal but a program is already running.

Remedy: Re-issue start request when current program is paused or aborted.

APSH-018 WARN Must complete fault recovery

Cause: The shell detected a run request but there is still fault recovery which has not been completed.

Remedy: Complete the fault recovery as directed under the Recovery menu (under ALARMS).

APSH-019 WARN Job root string is uninitialized

Cause: The shell could not build up the job name because \$SHELL_CFG.\$JOB_ROOT is uninitialized.

Remedy: Set \$SHELL_CFG.\$JOB_ROOT.

APSH-020 WARN Set \$SHELL_CFG.\$JOB_ROOT**APSH-021 WARN No paused program to continue**

Cause: A UOP CYCLE START input was detected but there is no paused program to continue.

Remedy: If UOP CYCLE START should run as well as resume, then set \$SHELL_CFG.\$CONT_ONLY=FALSE.

APSH-022 WARN \$SCR.\$CONT_ONLY = TRUE

CD Error Codes (ID = 82)**CD-001 WARN No global variables****Cause:** Coordinated Motion global variables are not loaded.**Remedy:** Perform a controlled start and initialize motion softparts.**CD-002 WARN Unable to allocate memory****Cause:** A failure occurred while allocating memory.**Remedy:** Check amount of memory being used by the system.**CD-003 STOP Follower recv invalid segment****Cause:** Leader segment MMR number does not match that of the follower.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

CD-004 STOP Illegal leader INTR point data**Cause:** Illegal Leader Interpolated Point Data is detected when trying to convert it to a transform.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

CD-005 STOP Non-coordinated group detected**Cause:** Coordinated Motion is used for a group which has not been SETUP for coordinated motion.**Remedy:** Check motion statement. Perform Coordinated Motion SETUP and perform a COLD START.**CD-006 STOP Illegal follower joint motion****Cause:** JOINT MOTYPE was used for a follower during coordinated motion.**Remedy:** Use LINEAR or CIRCULAR MOTYPE instead.**CD-007 STOP Circular motype not supported****Cause:** CIRCULAR MOTYPE not implemented.**Remedy:** Use LINEAR MOTYPE instead.**CD-008 STOP No leader****Cause:** There is no leader in the coordinated motion.**Remedy:** Check motion statement. Perform Coordinated Motion SETUP and then perform a COLD START.**CD-009 STOP More than one leader****Cause:** There is more than one leader in the coordinated motion.**Remedy:** Check motion statement. Perform Coordinated Motion SETUP and then perform a COLD START.**CD-010 STOP Invalid angle in point data****Cause:** Invalid Angle detected in Point Data.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

CD-011 STOP Error in flushing CD mailbox**Cause:** Error in reporting mailbox status.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

-
- CD-012 STOP Illegal leader motion**
Cause: Leader single group motion after coordinated motion not allowed.
Remedy: Issue non-coordinated motion involving the follower group.
- CD-013 WARN Jog group is not a leader**
Cause: Attempt to perform coordinated jog with a non-leader group.
Remedy: Select leader group for coordinated jog.
- CD-014 WARN Jog group has multi follower**
Cause: Attempt to perform coordinated jog with a leader group which has multiple followers.
Remedy: Select only one leader/follower pair.
- CD-015 STOP Wrist joint is not supported**
Cause: WRIST JOINT Motion is not supported with coordinated motion.
Remedy: Delete Wjnt motion instruction.
- CD-016 STOP INC motion is not supported**
Cause: INCREMENTAL motion is not supported with coordinated motion.
Remedy: Delete INC instruction.
- CD-017 STOP INDEP motn is not supported**
Cause: Independent motion is not supported with coordinated motion.
Remedy: Change Independent motion to simultaneous motion.
- CD-018 STOP No calibration for CD**
Cause: Calibration for coordinated motion is not done.
Remedy: Execute calibration of coordinated motion in SETUP screen.
- CD-019 STOP Illegal follower setting**
Cause: Number of follower is zero or two or greater on this motion.
Remedy: Set number of follower correctly or set group mask correctly.
- CD-020 WARN Not reach relative speed**
Cause: Follower can not reach relative speed in program.
Remedy: Teach follower and leader position again to reach relative speed.
- CD-021 STOP No kinematics in CD group**
Cause: Attempt to perform coordinated motion with non-kinematics robot.
Remedy: Initialize robot library correctly.
- CD-022 STOP Prev term type is not FINE**
Cause: Term type before coordinated motion is not Fine or CNT0.
Remedy: Change term type before coordinated motion to FINE or CNT0 or JOINT motion.
- CD-023 STOP Illegal CD setting**
Cause: Setting of coordinated motion is not correct.
Remedy: Check setting of coordinated motion in SETUP screen And set correctly.
- CD-024 WARN Calibration was inaccurate**
Cause: Teaching points is incorrect or Leader's mechanics is inaccurate.
Remedy: Check the mechanics and reteach the points.
- CD-026 STOP Illegal transition:nonCD<->CD**
Cause: An illegal transition (non CD->CD or CD -> non CD) has occurred.
Remedy: Add or remove the COORD motion option.
- CD-027 STOP Illegal follower transition**
Cause: A transition from one CD pair to another has occurred, but the same follower group is used in both CD pairs.
Remedy: Insert non-coordinated motion between coordinated motion of a different pair.

CMCC Error Codes (ID = 55)**CMCC-000 WARN unknown error (CM00)**

Cause: This is an internal system error.
Remedy: Perform a cold start on the controller.

CMCC-001 WARN CMC global variable failure

Cause: IntelliTrak system variables are not loaded.
Remedy: Perform a controlled start and initialize the motion softparts.

CMCC-002 WARN CMC data area not found

Cause: Program data area cannot be found.
Remedy: No action required.

CMCC-003 STOP CMC schedule no. not retrieved

Cause: IntelliTrak schedule number error.
Remedy: Select the program and use the DETAIL page to check or modify the schedule number within 0 to 3.

CMCC-004 STOP CMC error allocating data area

Cause: A system error occurred while allocating the internal data area.
Remedy: Perform a cold start on the controller.

CMCC-005 STOP CMC system variable failure

Cause: An IntelliTrak variable was not loaded.
Remedy: Perform a controlled start and initialize the motion softparts.

CMCC-006 STOP CMC illegal schedule number

Cause: The schedule number is not within the valid range of 0 to 3.
Remedy: Select a program and use the DETAIL page to set the schedule number correctly.

CMCC-007 STOP CMC illegal cmc type

Cause: \$cmc_type is not within the valid range of 0 to 2.
Remedy: Set \$cmsch[selected schedule num].\$cmc_type correctly.

CMCC-008 WARN Unsupported function code

Cause: This is an internal system error.
Remedy: Cold start the controller by powering off and then powering on while pressing the reset key. If the error is not cleared, document the events that led to the error and call the FANUC Robotics Hotline.

CMND Error Codes (ID = 42)

- CMND-001 WARN Directory not found**
Cause: The specified directory can not be found.
Remedy: Check the device and path that you entered. If none entered, check the system default device from the FILE Menu or from the KCL command, CHDIR.
- CMND-002 WARN File not found**
Cause: The specified file could not be found.
Remedy: Check to make sure the file has been spelled correctly and that it exists. Also verify the device and path name are correct.
- CMND-003 WARN File already exists**
Cause: The file already exists and could not be overwritten.
Remedy: Make sure the overwrite option has been specified.
- CMND-006 WARN Self copy not allowed**
Cause: A file cannot be copied to itself.
Remedy: Change the name of the destination file so it is different from the source file.
- CMND-010 WARN Source type code is invalid**
Cause: The source variable was not a position type when converting between a Cartesian and joint position.
Remedy: The valid position types are POSITION, JOINTPOS, XYZWPR, and XYZWPTEXT.
- CMND-011 WARN Destination type code is invalid**
Cause: The destination variable was not a position type when converting between a Cartesian and joint position.
Remedy: The valid position types are POSITION, JOINTPOS, XYZWPR, and XYZWPTEXT.
- CMND-012 WARN Type codes do not match**
Cause: The requested type code doesn't match the passed variable type.
Remedy: Internal error. Insure that the type code matches the variable type.
- CMND-013 WARN Representation mismatch**
Cause: An attempt was made to compare two positions that are not the same type.
Remedy: Both positions must be the same type. Convert one before comparing.
- CMND-014 WARN Positions are not the same**
Cause: Two positions were compared and found not to be equal.
Remedy: The two positions were not equal within the specified tolerance. This could be a normal occurrence. This warning is the logical opposite of SUCCESS.
- CMND-015 WARN Both arguments are zero**
Cause: Both arguments to ATAN2 were zero or an internal error occurred when attempting to convert a POSITION to XYZWPR.
Remedy: If calling ATAN2, insure that both arguments are not zero. If converting a POSITION, then it is not convertible to an XYZWPR.
- CMND-016 WARN Division by zero**
Cause: An attempt was made to divide by zero.
Remedy: This is an internal error. Insure that the divisor is not equal or close to zero.
- CMND-017 WARN Angle is out of range**
Cause: The rotational angle is too great.
Remedy: Insure that the rotational angle is no greater than 100 times PI, or about 314.15926...
- CMND-018 WARN Invalid device or path**
Cause: An invalid device or path has been specified.
Remedy: Check the device and path that you entered. If none entered, check the system default device from the FILE Menu or from the KCL command, CHDIR.
- CMND-019 WARN Operation cancelled**
Cause: The operation was cancelled because CTRL-C or CTRL-Y was pressed.
Remedy: None.
- CMND-020 WARN End of directory**
Cause: The directory listing is finished.
Remedy: None.

CMND-021 WARN Cannot rename file

Cause: The destination file name contained both alphanumeric characters and the global character '**.

Remedy: Use only alphanumeric characters or a single global character when renaming a file.

CNTR Error Codes

CNTR-001 WARN No global variables

Cause: Continuous Turn global variables are not loaded.
Remedy: Perform a controlled start and initialize motion softparts.

CNTR-002 WARN No MIR pointer

Cause: This is an internal system error.
Remedy: Perform a cold start on the controller.

CNTR-003 WARN WARN No sysvar pointer

Cause: This is an internal system error.
Remedy: Perform a cold start on the controller.

CNTR-004 WARN No cnir pointer

Cause: This is an internal system error.
Remedy: Perform a controlled start and initialize the motion softparts.

CNTR-005 WARN WARN Illegal Cont. Turn Axis

Cause: The continuous turn axis that was selected is not a valid cont. turn axis, or cn_gear_n1 or cn_gear_n2 have a zero value
Remedy: Check Continuous turn axis, cn_gear_n1, and cn_gear_n2 Select different continuous turn axis and/or set correct gear ratio for continuous turn axis.

CNTR-006 WARN WARN Unable to Allocate Memory

Cause: A failure occurred while allocating memory.
Remedy: Check amount of memory being used by the system.

CNTR-007 STOP.G STOP.G Serious Internal error (G:%d^2)

Cause: Internal Continuous Turn error
Remedy: Record error and report to hotline

CNTR-008 STOP.G Invalid dest. angle, (G:%d^2)

Cause: Invalid destination angle during linear motion. Incompatibility with Continuous Turn and other options.
Remedy: Check compatibility of motion options. Remove other options.

CNTR-009 WARN WARN Warn-Cont Vel too high(G:%d^2)

Cause: Continuous turn axis velocity is too high. cn_turn_no will not be valid because of high rotational speed.
Remedy: Lower contaxisvel. This warning may be ignored if cn_turn_no is not used.

CNTR-010 STOP.G STOP.G Ind.EV option not allowed.

Cause: Continuous turn is not compatible with independent extended axes The Ind.EV motion option is not allowed.
Remedy: Remove Ind.EV option or disable continuous turn on the group

CNTR-011 STOP.G STOP.G Axis speed exceeds lim(G:%d^2)

Cause: Programmed motion exceeds the speed limits on the continuous turn axis. Speed limit is 180 degrees per ITP time
Remedy: Lower the speed either through KAREL or Teach Pendant

CNTR-012 STOP.G STOP.G Ending Cont Rot on Rel Motion

Cause: Attempted to end Continuous Rotation with a Relative Motion
Remedy: Continuous Rotation must be ended with an absolute motion Use an absolute motion to end continuous rotation

COND Error Codes (ID = 4)**COND-001 WARN Condition does not exist****Cause:** Specified condition does not exist**Remedy:** Check for condition statements to verify if the specified condition has really been created or not.**COND-002 WARN Condition handler superseded****Cause:** The specified condition number already exists in the system, and has been superseded by the new condition.**Remedy:** This is just a notification, and you do not have to do anything for this warning message.**COND-003 WARN Already enabled, no change****Cause:** The specified condition is already enabled. No change has been made.**Remedy:** This is just a notification, and you do not have to do anything for this warning message.**COND-004 WARN Already disabled, no change****Cause:** The specified condition is already disabled. No change has been made.**Remedy:** This is just a notification, and you do not have to do anything for this warning message.**COND-005 WARN No more conditions defined****Cause:** No more conditions are defined for the specified task.**Remedy:** No action is required.**COND-009 WARN Break point encountered****Cause:** Break point has been encountered.**Remedy:** No action is required**COND-010 WARN Cond exists, not superseded****Cause:** Specified condition already exists. Condition was not superseded. This might indicate two condition handlers for the same task with the same condition handler.**Remedy:** Either renumber the condition handler or avoid re-defining the same condition handler.**COND-011 ABORT Scan time took too long****Cause:** There are too many conditions defined. It took too long to scan them all.**Remedy:** Reduce the number of conditions defined.

DICT Error Codes (ID = 33)**DICT-001 WARN Dictionary already loaded**

Cause: A dictionary cannot be reloaded if it was loaded into FROM.

Remedy: Load into a different language and use KCL SET LANG to set the language.

DICT-002 WARN Not enough memory to load dict

Cause: There is no more permanent memory available in the system to load another dictionary.

Remedy: Clear all unnecessary programs, dictionaries or variables.

DICT-003 WARN No dict found for language

Cause: There are no dictionaries loaded for the specified language.

Remedy: Use the DEFAULT language or a language in which a dictionary has been loaded.

DICT-004 WARN Dictionary not found

Cause: The specified dictionary was not found.

Remedy: Use KCL LOAD DICT to load the dictionary into the DEFAULT language or the current language.

DICT-005 WARN Dictionary element not found

Cause: The dictionary element was not found.

Remedy: Check the dictionary or element number to be sure it is specified correctly.

DICT-006 WARN Nested level too deep

Cause: Only five levels of dictionary elements can be nested.

Remedy: Fix the dictionary text file to include fewer nested levels.

DICT-007 WARN Dictionary not opened by task

Cause: The dictionary was never opened.

Remedy: Remove the close operation.

DICT-008 WARN Dictionary element truncated

Cause: The dictionary element was truncated because the KAREL string array is not large enough to hold all the data.

Remedy: Increase either the size of the string or the number of strings in the array.

DICT-009 WARN End of language list

Cause: The language list has completed.

Remedy: None.

DICT-010 WARN End of dictionary list

Cause: The dictionary list has completed.

Remedy: None.

DICT-011 WARN Dict opened by too many tasks

Cause: Only five dictionaries can be open by one task at one time.

Remedy: Load the dictionary to memory or close an unused dictionary.

DICT-012 WARN Low on FROM, loaded to memory

Cause: Not enough memory exists in FROM so the dictionary was loaded to CMOS for R-J2 and DRAM IMAGE for R-J2.

Remedy: None required.

DICT-013 WARN Cannot open dictionary file

Cause: The dictionary file does not exist on the specified device or in the specified directory.

Remedy: Select the proper device/directory and try again.

DICT-014 WARN Expecting DICT-0 in dictionary file

Cause: The dictionary text incorrectly specifies an element without a \$.

Remedy: Make sure all dictionary elements begin with \$.

DICT-015 WARN Reserved word not recognized

Cause: A reserved word was not recognized in the dictionary text.

Remedy: Check for misspelling or look up the correct word in the KAREL Reference Manual.

DICT-016 WARN Ending quote expected

Cause: The dictionary text incorrectly specifies an element without using quotes.

Remedy: Make sure all dictionary text is surrounded by double quotes. Use a backslash if you want an actual quote to appear in the text. For example, \"This is an example\" will produce "This is an example".

DICT-017 WARN Expecting element name or num

Cause: A reference to another element is expected.
Remedy: Use the element number to reference the element.

DICT-018 WARN Invalid cursor position

Cause: The cursor position is specified incorrectly or the values are outside the limits.
Remedy: Make sure the cursor position is valid. For example, use @1,1 for the first row and col respectively.

DICT-019 WARN ASCII character code expected

Cause: A series of digits are expected after the # to specify an ASCII character code.
Remedy: Remove the # or look up the ASCII character code in the KAREL Reference Manual.

DICT-020 WARN Reserved word expected

Cause: An identifier is expected after the & to specify a reserved word.
Remedy: Remove the & or look up the reserved word in the KAREL Reference Manual.

DICT-021 WARN Invalid character

Cause: An unexpected character was found in the dictionary text file.
Remedy: Make sure all dictionary text is correct.

DICT-022 WARN Dict already opened by task

Cause: The dictionary is already open by the task.
Remedy: None required.

DICT-023 WARN Dict does not need to be opened

Cause: Dictionaries loaded to memory do not need to be opened.
Remedy: Do not try to open the dictionary file.

DICT-024 WARN Cannot remove dictionary file

Cause: Dictionaries loaded to FROM cannot be removed or a dictionary cannot be removed if another task has it opened.
Remedy: Do not try to remove a dictionary loaded to FROM. Remove the dictionary from the same task which loaded it.

DICT-028 WARN No FROM write, loaded to memory

Cause: Not enough memory exists in FROM so the dictionary was loaded to CMOS for R-J2 and DRAM IMAGE for R-J2.
Remedy: None required.

DICT-029 WARN Help element not found

Cause: The help dictionary element was not found.
Remedy: Check the dictionary to be sure the help dictionary element was specified correctly. The help dictionary element must be specified with a question mark (?) followed by the element number.

DICT-030 WARN Function key element not found

Cause: The function key dictionary element was not found.
Remedy: Check the dictionary to be sure the function key element was specified correctly. The function key element must be specified with a caret (^) followed by the element number.

DICT-040 WARN Expecting element num after \$

Cause: The dictionary text incorrectly specifies an element number.
Remedy: Make sure all dictionary elements begin with \$ followed by the element number.

DICT-041 WARN Expecting element name after ,

Cause: The dictionary text incorrectly specifies an element name.
Remedy: Make sure all dictionary elements are specified as \,element_name\ after the add constant name.

DICT-042 WARN Expecting add constant name

Cause: The dictionary text was specified incorrectly.
Remedy: Make sure all dictionary elements are specified as "+add_const_name" after the element number.

DICT-043 WARN Element number out of sequence

Cause: The dictionary text was not specified in sequence.
Remedy: Make sure all dictionary elements are specified in sequential order.

DICT-044 WARN Warning - large hole in ele seq

Cause: The dictionary text has a large gap between element numbers.
Remedy: Reduce the gap in the element sequence. Each missing element uses up five bytes of memory.

DICT-045 WARN .LIT or .END mismatch

Cause: The dictionary text was specified incorrectly.
Remedy: Verify that each .LIT is matched with a .END.

DICT-046 WARN Command already encountered

Cause: The dictionary text was specified incorrectly.
Remedy: Remove the extra command.

DICT-047 WARN File extension required

Cause: The dictionary compressor expects a file extension.
Remedy: Use the .etx file extension for error text, the .utx file extension for uncompressed text, or the .ftx file extension for form text.

DICT-048 WARN Invalid file extension

Cause: The dictionary compressor did not recognize the file extension.
Remedy: Use the .etx file extension for error text, the .utx file extension for uncompressed text, or the .ftx file extension for form text.

DICT-049 WARN Expecting file name

Cause: The dictionary compressor expects a file name.
Remedy: Specify a file name after the command.

DICT-050 WARN Expecting facility number

Cause: The dictionary compressor expects a facility number in the .KL command.
Remedy: Specify the facility number after the file name.

DICT-051 WARN Symbol invalid for dictionary type

Cause: An invalid command was specified for this type of dictionary file.
Remedy: Check the command and if a form is used, verify the file extension is .ftx.

DICT-052 WARN Expecting .ENDFORM symbol

Cause: The dictionary text was specified incorrectly.
Remedy: Verify that each .FORM is matched with a .ENDFORM.

DICT-053 WARN Cannot open include file

Cause: The include file could not be created.
Remedy: Make sure a valid file name has been specified.

DICT-054 WARN Form is being displayed

Cause: The form you are trying to compress is currently being displayed.
Remedy: Abort the KAREL program that is displaying the form.

DNET (DeviceNet) Error Codes

DNET-001 STOP No system device file

Cause: The system device definition file is missing from the system.

Remedy: INIT start and reload the DeviceNet Interface option. If the error still exists, document the events that led to the error and call your FANUC Robotics technical representative.

DNET-002 STOP No application device file

Cause: The application device definition file is missing from the system.

Remedy: INIT start and reload the DeviceNet Interface option. If the error still exists, document the events that led to the error and call your FANUC Robotics technical representative.

DNET-004 STOP Board init failed: Bd %d

Cause: The specified board has failed to initialize.

Remedy: Make sure the board parameters are correct. Make sure the board is properly connected to the network and power is supplied.

DNET-006 ERR_SYS_C System error: %d

Cause: A system error has occurred.

Remedy: Document the events that led to the error and call your FANUC Robotics technical representative.

DNET-008 STOP Invalid board index

Cause: An invalid board index has been specified.

Remedy: Specify a board index between 0 and 3.

DNET-009 STOP Invalid MAC Id: Bd %d MAC %d

Cause: An invalid MAC Id has been specified.

Remedy: Specify a MAC Id between 0 and 63 inclusive.

DNET-010 STOP Board already online

Cause: The specified board is already on-line.

Remedy: Take the board off-line before attempting the operation.

DNET-011 STOP Board not online

Cause: The specified board is not on-line.

Remedy: Put the board on-line before attempting the operation.

DNET-012 STOP Device already online

Cause: The specified device is already on-line.

Remedy: Take the device off-line before attempting the operation.

DNET-013 STOP Device not online

Cause: The specified device is not on-line.

Remedy: Put the device on-line before attempting the operation.

DNET-014 STOP Request timed out

Cause: The attempted DeviceNet command request has timed out.

Remedy: Check all network connections. If all connections appear to be in order, re-attempt the command.

DNET-015 STOP Board not initialized

Cause: The specified board has not been initialized.

Remedy: Initialize the board by attempting to put it on-line, and then cycle power. Then, re-attempt the operation.

DNET-016 STOP System failed

Cause: The DeviceNet Interface system has failed.

Remedy: Cold start the system. If the problem persists, INIT start or reload the system. If the problem continues to persist, document the events that led to the error and call your FANUC Robotics technical representative.

DNET-017 STOP Board not found

Cause: The specified board was not found in the system.

Remedy: Make sure the daughter boards are properly configured and properly seated on the motherboard.

DNET-018 STOP Memory test failed

Cause: The specified board has failed the initial memory test.

Remedy: Cold start the system. If the problem persists, INIT start and reload the DeviceNet Interface option.

DNET-019 STOP Code file open failed

Cause: The code file required to initialize the board cannot be accessed.

Remedy: Cold start the system. If the problem persists, INIT start and reload the DeviceNet Interface option.

DNET-020 STOP Code file read failed

Cause: The code file required to initialize the board cannot be read.

Remedy: Cold start the system. If the problem persists, INIT start and reload the DeviceNet Interface option.

DNET-021 STOP Code file checksum error

Cause: There is a problem with the DeviceNet scanner code file.

Remedy: Cold start the system. If the problem persists, INIT start and reload the DeviceNet Interface option. If the problem continues to persist, document the events that led to the error and call your FANUC Robotics technical representative.

DNET-022 STOP Board initialization timeout

Cause: The board initialization routine has timed out.

Remedy: Turn the controller off. Make sure the motherboard is correctly seated on the back plane. Cold start the controller. If the problem persists, document the events that led to the error and call your FANUC Robotics technical representative.

DNET-023 STOP Board initialization error

Cause: An error has occurred in the board initialization process.

Remedy: Cycle power to the controller. If the problem persists, turn the controller off and check the motherboard connection to the back plane. Cold start the controller. If the problem persists, document the events that led to the error and call your FANUC Robotics technical representative.

DNET-025 STOP No device assigned for Bd/MAC

Cause: A data mismatch has occurred such that the system cannot find a device assigned for the specified board number and MAC Id.

Remedy: Turn the controller off and cold start the controller. If the problem persists, delete the board from the Board List screen, reconfigure the board, and re-add devices to the Device List. Cycle power. Also, check the device MAC Id configurations.

DNET-026 STOP No match on dev type look-up

Cause: The system cannot find the specified device type in its list of defined device types.

Remedy: Check the selected device type on the Device List. Next, check the Defined Device List and the Standard Device Definition List for the required device type. If it does not appear, go to the Defined Device List and add the required device definition, then select it on the Device List screen. When you have finished, turn off then turn on the controller.

DNET-027 STOP Dev online err: Bd %d MAC %d

Cause: The device at the specified board number and MAC Id cannot be brought on-line.

Remedy: Make sure the device is properly connected to the network. Check the device's MAC Id and baud rate configuration. Check the board's baud rate configuration on the Board Detail screen. Check the board's network connection. Cold Start the controller.

DNET-028 STOP Board online err: Bd %d

Cause: The specified board cannot be brought on-line.

Remedy: Make sure the board is properly connected to the network. Check that network power is being supplied. Check that baud rates for the board and devices are in agreement. Cold Start the controller.

DNET-029 STOP Too many deferred errors

Cause: The system has received the maximum number of DeviceNet errors it can handle at one time.

Remedy: Attempt to remedy any errors that are displayed, then cold start the controller.

DNET-030 STOP Std dev file fmt err: Line %d

Cause: There is an error in the format of the specified device definition file, on the specified line.

Remedy: Contact your FANUC Robotics technical representative to obtain a correct device definition file.

DNET-031 STOP App dev file fmt err: Line %d

Cause: There is an error in the format of the specified device definition file, on the specified line.

Remedy: Contact your FANUC Robotics technical representative to obtain a correct device definition file.

DNET-033 STOP Unknown keyword

Cause: An unknown keyword has been found in the device definition files.

Remedy: The DeviceNet option software is corrupted. Re-install the DeviceNet Interface option. If the problem persists, contact FANUC Robotics to obtain a new Communication Options software disk.

DNET-035 STOP Bad format or out of range

Cause: An integer value in the device definition files is incorrect.

Remedy: The DeviceNet option software is corrupted. Re-install the DeviceNet Interface option. If the problem persists, contact FANUC Robotics to obtain a new Communication Options software disk.

DNET-036 STOP No NINPUTS or NOUTPUTS line

Cause: The specified line was not found in a device definition file.

Remedy: The DeviceNet option software is corrupted. Re-install the DeviceNet Interface option. If the problem persists, contact FANUC Robotics to obtain a new Communication Options software disk.

DNET-037 STOP No PDTCODE line

Cause: The specified line was not found in a device definition file.

Remedy: The DeviceNet option software is corrupted. Re-install the DeviceNet Interface option. If the problem persists, contact FANUC Robotics to obtain a new Communication Options software disk.

DNET-038 STOP No MODULE lines with MULTIMOD

Cause: The specified lines were not found in a device definition file.

Remedy: The DeviceNet option software is corrupted. Re-install the DeviceNet Interface option. If the problem persists, contact FANUC Robotics to obtain a new Communication Options software disk.

DNET-039 STOP Too many MODULE lines

Cause: The specified lines were incorrect in a device definition file.

Remedy: The DeviceNet option software is corrupted. Re-install the DeviceNet Interface option. If the problem persists, contact FANUC Robotics to obtain a new Communication Options software disk.

DNET-040 STOP MODULE specified w/o MULTIMOD

Cause: A definition was incorrect in a device definition file.

Remedy: The DeviceNet option software is corrupted. Re-install the DeviceNet Interface option. If the problem persists, contact FANUC Robotics to obtain a new Communication Options software disk.

DNET-041 STOP Required field missing

Cause: A definition was incorrect in a device definition file.

Remedy: The DeviceNet option software is corrupted. Re-install the DeviceNet Interface option. If the problem persists, contact FANUC Robotics to obtain a new Communication Options software disk.

DNET-042 STOP No DEVTYPE line supplied

Cause: The specified line was not found in a device definition file.

Remedy: The DeviceNet option software is corrupted. Re-install the DeviceNet Interface option. If the problem persists, contact FANUC Robotics to obtain a new Communication Options software disk.

DNET-043 STOP No VENDORID line supplied

Cause: The specified line was not found in a device definition file.

Remedy: The DeviceNet option software is corrupted. Re-install the DeviceNet Interface option. If the problem persists, contact FANUC Robotics to obtain a new Communication Options software disk.

DNET-044 STOP No PRODCODE line supplied

Cause: The specified line was not found in a device definition file.

Remedy: The DeviceNet option software is corrupted. Re-install the DeviceNet Interface option. If the problem persists, contact FANUC Robotics to obtain a new Communication Options software disk.

DNET-045 STOP No I/O type line supplied

Cause: The specified line was not found in a device definition file.

Remedy: The DeviceNet option software is corrupted. Re-install the DeviceNet Interface option. If the problem persists, contact FANUC Robotics to obtain a new Communication Options software disk.

DNET-046 STOP No PDTCODE line supplied

Cause: The specified line was not found in a device definition file.

Remedy: Contact your FANUC Robotics technical representative to obtain the correct device definition files.

DNET-047 STOP DeviceNet motherboard not found

Cause: The DeviceNet motherboard is not plugged into the back plane.

Remedy: Turn off the controller and make sure the motherboard is properly seated into the back plane of the controller. Cold start the controller.

DNET-052 STOP Data line too long

Cause: The specified line was incorrect in the device definition file.

Remedy: The DeviceNet option software is corrupted. Re-install the DeviceNet Interface option. If the problem persists, contact FANUC Robotics to obtain a new Communication Options software disk.

DNET-053 STOP Line above DEVICE line ignored

Cause: An extraneous line was found in a device definition file.

Remedy: Check the Standard Device Definition List to see if device types have been properly loaded. If not, contact FANUC Robotics to obtain the correct device definition files.

DNET-054 STOP All space in shared RAM used

Cause: There is no space left in the DeviceNet I/O buffer.

Remedy: Contact FANUC Robotics to report the problem. Provide all details of the DeviceNet network, including number and type of devices, baud rates, MAC Ids, and network wiring configuration.

DNET-055 STOP Board or network error: Bd %d

Cause: An error has occurred with the specified daughter board or the DeviceNet network connected to it.

Remedy: Refer to the next DNET alarm posted in the alarm log for specific alarm recovery information.

DNET-056 STOP Network power lost

Cause: Power has been removed from the DeviceNet network.

Remedy: Check the cable connecting the daughter board to the DeviceNet network. Also, check the connection to the power source. Cycle power to the controller.

DNET-057 STOP Network communications error

Cause: A network communications error has occurred on the network connected to the specified board.

Remedy: Check that the board's baud rate corresponds to that of the devices. Check cable connections to both the board and devices. Check that the proper device definitions are selected for the devices on the network and that parameters are correct for user-defined devices. Turn off both the controller and the DeviceNet network power, then cold start the controller.

DNET-058 STOP Message queue overrun

Cause: The board has received more messages than it can handle at one time.

Remedy: The problem may be momentary; attempt to bring the board on-line again. If the problem persists, check that the board baud rate corresponds to the baud rate of the devices. Turn off then turn on the controller.

DNET-059 STOP Message lost

Cause: The board has missed a message over the DeviceNet network.

Remedy: The problem might be momentary; attempt to bring the board on-line again. If the problem persists, check that the board baud rate corresponds to the baud rate of the devices. Turn off then turn on power to the controller.

DNET-060 STOP Xmit timeout: Network flooded

Cause: The traffic on the DeviceNet network is too heavy for the board to communicate with the devices.

Remedy: Check that the board baud rate agrees with the baud rate of the devices. If no baud rate problem exists, turn off both the controller and the DeviceNet network power, then turn on both.

DNET-061 STOP No other nodes on network

Cause: All of the devices expected by the board to be on the network appear to be disconnected to the network.

Remedy: Check cable connections to the board and to the devices. If a device has been disconnected, reconnect and press RESET on the teach pendant. Check that the board baud rate is the same as baud rate of the devices.

DNET-062 STOP Bus off due to comm errors

Cause: The board is not communicating to the network because there are too many errors.

Remedy: Check that the baud rate of the board and of the devices is the same. Make sure that power is connected to the DeviceNet network. Press RESET on the teach pendant. If the problem persists, begin removing devices from the network; after each device is removed, press RESET. When the board is brought on-line, check the device configuration and the parameters of the device definition.

DNET-063 STOP Device error: Bd %d MAC %d

Cause: An error has occurred with the device at the specified board number and MAC Id.

Remedy: Refer to the next DNET alarm posted in the alarm log for specific alarm recovery information.

DNET-064 STOP Connection error

Cause: An error has occurred when attempting connection to the specified device.

Remedy: Check that the baud rate of the device agrees with the board baud rate. Check that the device is properly connected to the network; make sure the device is receiving power from the network. Inspect the device definition to see that the I/O type, access mode, and size of I/O are correct. Press RESET on the teach pendant to re-attempt connection.

DNET-065 STOP Incorrect vendor Id

- Cause:** The vendor Id for the device, as specified in the device definition, is incorrect.
Remedy: Delete the device from the Device List. Check the device documentation for the correct vendor Id. Make corrections in the device definition and add the device to the Device List.

DNET-066 STOP Incorrect product code

- Cause:** The product code for the device, as specified in the device definition, is incorrect.
Remedy: Delete the device from the Device List. Check the device documentation for the correct product code. Make corrections in the device definition and add the device to the Device List.

DNET-067 STOP Incorrect device type

- Cause:** The device type for the device, as specified in the device definition, is incorrect.
Remedy: Delete the device from the Device List. Check the device documentation for the correct device type. Make corrections in the device definition and add the device to the Device List.

DNET-068 STOP Device timeout

- Cause:** The connection to the specified device has timed out.
Remedy: Check the device's connection to the network. Make sure the device baud rate agrees with the board baud rate. Attempt to bring the device on-line by pressing RESET on the teach pendant.

DNET-069 STOP Unknown error code %d

- Cause:** An unknown error has occurred with the specified device.
Remedy: Document the events that led to the error and call your FANUC Robotics technical representative. Make sure the error code number is noted and reported.

DNET-073 STOP No match on mod type look-up

- Cause:** The system could not find the module type corresponding to a module on the specified device.
Remedy: View the module list for the device and delete or change the module in question. If this module was previously functional, cold start the controller and attempt to use this module type again. If the problem persists, perform an INIT start and re-load the DeviceNet Interface option.

DNET-074 STOP Load only at ctrl start

- Cause:** An I/O configuration file (.IO file) containing DeviceNet configuration data was loaded at COLD START. The DeviceNet configuration data in this file is ignored.
Remedy: Reload the .IO file at controlled start.

DNET-076 STOP \$DN_DEV_DEFS array is full

- Cause:** There is no more room in the Defined Device List system variable.
Remedy: Delete any unneeded device definitions from the Defined Device List before adding a new one.

DNET-078 STOP No room for more devices

- Cause:** The system variable for storage of devices is full.
Remedy: If there are devices which are off-line, delete these devices unless they are required to be kept on the Device List. After entries in the device list are freed, new devices can be added.

DNET-079 STOP Unknown dev type: Bd %d MAC %d

- Cause:** The device type used by this device is currently unknown to the system.
Remedy: This error occurs during the I/O restore. Cold start the controller, add a new device definition corresponding to the specified device, then add the device to the device list.

DNET-080 STOP Loaded config too large

- Cause:** The previous I/O configuration contains too many modules, devices, or device definitions to be loaded.
Remedy: Make sure you have the same memory configuration as the system on which the I/O configuration was saved.

DNET-084 STOP Board reset failed: Bd %d

- Cause:** The command to reset the specified board has failed.
Remedy: Refer to the next DNET alarm posted in the alarm log for specific alarm recovery information.

DNET-085 STOP Dev reset failed Bd %d MAC %d

- Cause:** The command to reset the specified device has failed.
Remedy: Refer to the next DNET alarm posted in the alarm log for specific alarm recovery information.

DNET-086 STOP Stop scan cmd failed: Bd %d

- Cause:** The specified board is unable to acknowledge the stop-scanning command.
Remedy: Check DeviceNet connection to the board, as well as DeviceNet power to the network. If board is already in ERROR state, this error can be disregarded.

DNET-087 STOP Bd offline cmd failed: Bd %d

Cause: The board is not acknowledging the command to take it off-line.

Remedy: Check DeviceNet connection to the board, as well as DeviceNet power to the network. If the board is already in the ERROR state, this error can be disregarded.

DNET-088 STOP Ignored: Bd %d MAC %d Slot %d

Cause: The system does not recognize the module type of the module being loaded.

Remedy: Make sure the device definition data files are the same between the current system being loaded and the system on which the I/O configuration was saved. Contact FANUC Robotics for the correct definition files.

DNET-089 STOP Can't specify POLL and STROBE

Cause: The data file contains lines which specify both POLL access and STROBE access for the same device.

Remedy: The DeviceNet option software is corrupted. Re-install the DeviceNet Interface option. If the problem persists, contact FANUC Robotics to obtain a new Communication Options software disk.

DNET-090 STOP Can't STROBE w/ num outs > 0

Cause: The device definition file specifies a strobed-access device but the number of outputs is not equal to zero.

Remedy: The DeviceNet option software is corrupted. Re-install the DeviceNet Interface option. If the problem persists, contact FANUC Robotics to obtain a new Communication Options software disk.

DNET-091 STOP Input size error

Cause: The number of inputs specified in the device definition for this device does not match the number expected by the scanner when it communicates with the device.

Remedy: Delete the device, correct the device definition, then re-add the device to the device list.

DNET-092 STOP Output size error

Cause: The number of outputs specified in the device definition for this device does not match the number expected by the scanner when it communicates with the device.

Remedy: Delete the device, correct the device definition, then re-add the device to the device list.

DNET-093 STOP Error reading vendor ID

Cause: The scanner board encountered an error while trying to read the device's vendor ID.

Remedy: Check that the device baud rate matches the board baud rate. Check also the device's connection to the network.

DNET-094 STOP Error reading device type

Cause: The scanner board encountered an error while trying to read the device's device type.

Remedy: Check that the device baud rate matches the board baud rate. Check also the device's connection to the network.

DNET-095 STOP Error reading product code

Cause: The scanner board encountered an error while trying to read the device's product code.

Remedy: Check that the device baud rate matches the board baud rate. Check also the device's connection to the network.

DNET-096 STOP Error setting packet rate

Cause: The scanner board encountered an error while trying to set the communication packet rate for this device.

Remedy: Check that the device baud rate matches the board baud rate. Check also the device's connection to the network. Reset the device if possible.

DNET-097 STOP Connection sync fault

Cause: The board was unable to achieve synchronization in the connection with the specified device.

Remedy: Check that the device baud rate matches the board baud rate. Check also the device's connection to the network. Reset the device if possible.

DNET-102 STOP Invalid board MAC Id

Cause: The board's MAC Id is not between 0 and 63.

Remedy: Check the Board Detail screen to see if the board's MAC Id is between 0 – 63, inclusive. If it is not, change the MAC Id to a valid value and press RESET on the teach pendant. If the MAC Id appears valid, cold start the controller. If the problem persists, document the events that led to the error and call your FANUC Robotics technical representative.

DNET-103 STOP Invalid board baud rate

Cause: The board's baud rate is not one of: 125 KB, 250 KB, or 500 KB.

Remedy: Check the Board Detail screen to see if the board's baud rate is one of the above values. If it is not, change the baud rate to a valid value and press RESET on the teach pendant. If the baud rate appears valid, cold start the controller. If the problem persists, document the events that led to the error and call your FANUC Robotics technical representative.

DNET-104 STOP Duplicate MAC Id error

Cause: The specified device has the same MAC Id as another device on the network.

Remedy: Check that no other devices have the same MAC Id, particularly those connected to a different master on the same network. Change the MAC Id of the offending device at both the device and on the Device List, and attempt to bring it on-line. If the problem persists, cold start the controller and try again. If the problem continues, document the events that led to the error and call your FANUC Robotics technical representative.

DNET-105 STOP Duplicate device error

Cause: There was an attempt to add a device to the board's device list that was a duplicate of a device already on the list.

Remedy: If the desired device is already on the network and a second one is not being added, you may ignore the error. Otherwise, change the MAC Id of one of the duplicate devices.

DNET-106 STOP Device not found error

Cause: A device expected to be on the network was not found.

Remedy: Check that the device is connected to the network. Check that the device baud rate matches the board baud rate. Reset the device if possible. Cycle power to the controller. If the problem persists, document the events that led to the error and call your FANUC Robotics technical representative.

DNET-107 STOP Bus offline error

Cause: The board could not perform an operation because the bus was off-line.

Remedy: Press RESET on the teach pendant to attempt to bring the board on-line. If the problem persists, cycle power to the controller. If the problem continues to persist, cycle power to the DeviceNet network.

DNET-108 STOP Scanner active error

Cause: The board could not perform an operation because the it is actively scanning the network.

Remedy: Take the board off-line and re-attempt the operation.

DNET-109 STOP Bus not offline error

Cause: The board could not perform an operation because the bus is not off-line.

Remedy: Take the board off-line and re-attempt the operation.

DNET-110 STOP Error: board scanning

Cause: The board could not perform an operation because the it is actively scanning the network.

Remedy: Take the board off-line and re-attempt the operation.

DNET-111 STOP Error: board not scanning

Cause: The board could not perform an operation because the it is not actively scanning the network.

Remedy: Bring the board on-line and re-attempt the operation.

DNET-112 STOP Board not ready; pls. wait

Cause: An attempt to bring the board on-line was unsuccessful because the board was busy.

Remedy: Wait ten seconds and re-attempt to bring the board on-line. If the problem persists, check board connection to the network, baud rate, and network power.

DNET-114 STOP Bus fault error detected

Cause: The board has detected a fault on the DeviceNet network, and cannot communicate with devices.

Remedy: Check that the baud rate of the board matches the baud rate of all devices on the network. Also, check that power is being supplied to the network. If the problem persists, cycle power to the controller, and then to the network if the problem continues.

ELOG Error Codes (ID = 5)**ELOG-009 WARN call a service man****Cause:** A system error has occurred.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

ELOG-011 WARN Power off, if you want to recover.**Cause:** A system error has occurred.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

ELOG-012 WARN A system error has been occurred.**Cause:** A system error has occurred**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

FILE Error Codes (ID = 2)

FILE-001 WARN Device not ready

Cause: Specified file device is not ready.

Remedy: Check if the device is mounted and ready to use. Check if the device name is correct.

FILE-002 WARN Device is Full

Cause: Device is full. There is no more space to store data on the device.

Remedy: Delete any unnecessary files or change to a new device.

FILE-003 WARN Device is protected

Cause: Device is protected. So, you cannot write to the device.

Remedy: Release the device protection.

FILE-005 WARN Device not mounted

Cause: Device is not mounted. You should mount the device before using it.

Remedy: Mount the correct file device.

FILE-006 WARN Device is already mounted

Cause: You tried to mount the device which had been already mounted.

Remedy: Mount device only once.

FILE-008 WARN Illegal device name

Cause: Device name contains an illegal character.

Remedy: Check spelling and validity of device name.

FILE-009 WARN Illegal logical unit number

Cause: Illegal LUN is used.

Remedy: This is an internal error. Check the validity of the logical unit number.

FILE-010 WARN Directory not found

Cause: Specified directory does not exist

Remedy: Check validity of directory name.

FILE-011 WARN Directory full

Cause: Directory is full. You tried to create a file in the root directory which exceeded the maximum number of files allowed on the device.

Remedy: Delete unnecessary files in the root directory.

FILE-012 WARN Directory is protected

Cause: You tried to write to a write protected directory.

Remedy: Release the protection to the directory.

FILE-013 WARN Illegal directory name

Cause: Directory name contains an illegal character.

Remedy: Check spelling of directory name.

FILE-014 WARN File not found

Cause: The specified file was not found.

Remedy: Check that the file exists and that the file name was spelled correctly.

FILE-015 WARN File is protected

Cause: You tried to access a protected file.

Remedy: Release the protection from file.

FILE-017 WARN File not open

Cause: You tried to access a file which is not open.

Remedy: Open the file before accessing.

FILE-018 WARN File is already opened

Cause: You tried to create/delete/rename a file which is already opened.

Remedy: Close file before such operations.

FILE-019 WARN File is locked

Cause: You tried to access a file which is locked.

Remedy: Release the lock.

FILE-020 WARN Illegal file size**Cause:** File size is invalid.**Remedy:** Change file size to be correct.**FILE-021 WARN End of file****Cause:** End of file was detected.**Remedy:** This is a notification. You do not have to do anything for this warning message.**FILE-022 WARN Illegal file name****Cause:** File name contains an illegal character.**Remedy:** Check spelling of file name.**FILE-023 WARN Illegal file number****Cause:** File number is illegal.**Remedy:** Use a valid file number which is the ID returned from an open request.**FILE-024 WARN Illegal file type****Cause:** File type contains an illegal character.**Remedy:** Check the spelling and validity of the file type.**FILE-025 WARN Illegal protection code****Cause:** File protection code is illegal.**Remedy:** Check if the protection code is correct.**FILE-026 WARN Illegal access mode****Cause:** File access mode is illegal.**Remedy:** Check if the access mode is correct.**FILE-027 WARN Illegal attribute****Cause:** File attribute in the SET_ATTRIBUTE request is illegal.**Remedy:** Check that attribute specified is valid.**FILE-028 WARN Illegal data block****Cause:** Data block is broken which is used in FIND_NEXT request.**Remedy:** You should keep the data block which is returned from the previous FIND_FIRST or FIND_NEXT request.**FILE-029 WARN Command is not supported****Cause:** Illegal request command is specified.**Remedy:** Check if the request code is correct.**FILE-030 WARN Device lun table is full****Cause:** Device management table is full.**Remedy:** Dismount any unnecessary devices.**FILE-031 WARN Illegal path name****Cause:** Path name contains an illegal character.**Remedy:** Check if the path name is correct.**FILE-032 WARN Illegal parameter****Cause:** Illegal parameter is detected.**Remedy:** Check that all parameters for the request are valid.**FILE-033 WARN System file buffer full****Cause:** File management buffer is full.**Remedy:** Close unnecessary files.**FILE-034 WARN Illegal file position****Cause:** Illegal file position is specified.**Remedy:** Check that the file position parameter from SEEK request is positive and not beyond the end of file.**FILE-035 WARN Device not formatted****Cause:** You tried to access a unformatted device.**Remedy:** Format the device before using it.**FILE-036 WARN File already exist****Cause:** You tried to rename a file to an already existing file name.**Remedy:** Change the new file name to be unique or delete the existing file.

FILE-037 WARN Directory not empty

Cause: You tried to remove a subdirectory which contains some files or directories.

Remedy: Remove all files and directories in the subdirectory before removing subdirectory.

FILE-038 WARN File locked by too many tasks

Cause: There are too many lock requests to same file.

Remedy: Unlock any unnecessary file lock requests.

FILE-039 WARN Directory already exists

Cause: You tried to create a sub-directory that already exists.

Remedy: Use a unique name for new sub-directory

FILE-040 WARN Illegal file access mode

Cause: You tried to read from a write only opened file or tried to write to a read only opened file.

Remedy: Open a file with correct access mode.

FILE-041 WARN File not locked

Cause: You tried to unlock file which you had not locked.

Remedy: Don't unlock a file that is not locked. You can only unlock files which YOU have locked.

FILE-045 WARN need to set \$FILE_MAXSEC

Cause: \$FILE_MAXSEC has not been set and must be set before device can be formatted.

Remedy: Set variable \$FILE_MAXSEC to valid value. 800 is a good default value.

FLPY Error Codes (ID = 10)

FLPY-001 WARN End of directory reached

Cause: Your listing has reached the end of the directory. You do not have to do anything for this warning message.

Remedy: This is a notification. You do not have to do anything for this warning message.

FLPY-002 WARN File already exists

Cause: The file name you are trying to create already exists on this device.

Remedy: Delete the file of this name or choose a different file name.

FLPY-003 WARN File does not exist

Cause: The file you are trying to open does not exist on this device.

Remedy: Open a file that does exist on the device.

FLPY-004 WARN Unsupported command

Cause: Operation is not supported on floppy disk.

Remedy: Use only operations supported on floppy disk.

FLPY-005 WARN Disk is full

Cause: The disk file capacity has been reached.

Remedy: Delete some unneeded files or use a disk with sufficient free space.

FLPY-006 WARN End of file reached

Cause: The end of the file was reached while reading.

Remedy: Do not attempt to read beyond the end of a file.

FLPY-008 WARN Only one file may be opened

Cause: An attempt was made to open more than one file.

Remedy: Do not attempt to open more than one file at a time.

FLPY-009 WARN Communications error

Cause: The protocol format was invalid.

Remedy: Retry the operation.

FLPY-015 WARN Write protection violation

Cause: The disk has write protection enabled.

Remedy: Remove write protection from the disk or use a disk that is not write protected.

FLPY-100 WARN Directory read error

Cause: The directory information is corrupted and unreadable.

Remedy: Try another disk or reformat the disk.

FLPY-101 WARN Block check error

Cause: The checksum data is bad. Data is corrupted on disk and can not be read.

Remedy: Try another disk, or reformat the disk

FLPY-103 WARN Seek error

Cause: There is a bad sector or track on the disk.

Remedy: Clean the disk drive, try another disk, or reformat the disk.

FLPY-104 WARN Disk timeout

Cause: The drive did not respond to a command.

Remedy: Check the cable to the drive and make sure drive power is on.

FLPY-105 WARN Write protection violation

Cause: The disk has write protection enabled.

Remedy: Remove write protection from the disk or use a disk that is not write protected.

FRSY Error Codes (ID = 85)

FRSY-001 WARN FROM disk is full

- Cause:** The FROM disk does not have enough available memory to perform the specified command.
Remedy: Delete all unnecessary files and then purge the device. If the device is still full, then backup the files to an off-line device and reformat the device.

FRSY-002 WARN Device not formatted

- Cause:** The device is not formatted.
Remedy: Format the device before using it.

FRSY-003 WARN Invalid parameter

- Cause:** An invalid parameter is detected.
Remedy: Verify all the parameters for the requested command are correct.

FRSY-004 WARN RAM disk must be mounted

- Cause:** Copying a file to the FROM disk requires that the RAM disk be mounted with enough memory available to temporarily contain the file.
Remedy: Mount the RAM disk before specifying the command.

FRSY-005 WARN Device not mounted

- Cause:** The device is not mounted.
Remedy: Mount the device before using it.

FRSY-006 WARN Device is already mounted

- Cause:** The device is already mounted.
Remedy: This is a notification. You do not have to do anything for this warning message.

FRSY-007 WARN Invalid device name

- Cause:** The specified device is not valid.
Remedy: Verify the device name.

FRSY-008 WARN File already exists

- Cause:** The specified file already exists.
Remedy: Delete the file first or specify overwrite if available with the command.

FRSY-009 WARN Too many files opened

- Cause:** The maximum number of files is already open. Therefore the requested command cannot be performed.
Remedy: Either close one or more of the files or set \$OPEN_FILES to a larger number and perform a cold start.

FRSY-010 WARN Invalid file position

- Cause:** An invalid file position is specified. The position is beyond the end of the file or a negative position.
Remedy: Check the file position.

FRSY-011 WARN Directory full

- Cause:** No more files are allowed on the device.
Remedy: Delete any unnecessary files or dismount and remount MF: device which will increase the maximum number of files allowed.

FRSY-012 WARN Invalid file access mode

- Cause:** The requested command cannot be performed because the file is not opened with the proper access mode. This error is also caused by trying to update or append to an existing file on the FROM disk or to an existing compressed file on the RAM disk. Update and append are only allowed with uncompressed files on the RAM disk.
Remedy: Open the file with the proper access mode.

FRSY-013 WARN Device is too fragmented

- Cause:** The file cannot be created on the device because not enough consecutive blocks are available.
Remedy: Delete all unnecessary files and then purge the device. For more information on purging, refer to the PURGE_DEV Built-in in the *FANUC Robotics SYSTEM R-J2 KAREL Reference Manual*. If the device is still full, then backup the files to an off-line device and reformat the device.

FRSY-014 WARN File not found

- Cause:** The specified file is not found.
Remedy: Verify the file name and the specified or default device is correct.

FRSY-015 WARN Invalid file name

- Cause:** The file name contains an invalid character or is blank.
Remedy: Verify the file name is correct.

FRSY-016 WARN Invalid file type

Cause: The file type contains an invalid character.
Remedy: Verify the file type is correct.

FRSY-017 WARN File not open File not open

Cause: The file is not open.
Remedy: Open the file before accessing.

FRSY-018 WARN File is already opened

Cause: The requested command cannot be performed because the file is already opened.
Remedy: Close the file before specifying the command.

FRSY-019 WARN Command is not supported

Cause: The specified command is not supported for the device.
Remedy: This is a notification. You do not have to do anything for this warning message.

FRSY-020 WARN RAM disk is full

Cause: The RAM disk does not have enough available memory to perform the specified command. Note that copying a file to the FROM disk requires that the RAM disk be mounted with enough memory available to temporarily contain the file.
Remedy: Delete all unnecessary files and then purge the device. For more information on purging, refer to the PURGE_DEV Built-in in the *FANUC Robotics SYSTEM R-J2 KAREL Reference Manual*. If the device is still full, then backup the files to an off-line device and reformat the device after setting \$FILE_MAXSEC to a larger number.

FRSY-021 WARN End of file

Cause: The end of the file is detected.
Remedy: This is a notification. You do not have to do anything for this warning message.

FRSY-022 WARN File ID exceeded maximum

Cause: The file identification number has reached the maximum number for the device.
Remedy: You must backup all your files, reformat the device, and restore the files. Refer to chapter 9, "File System", in the *FANUC Robotics SYSTEM R-J2 KAREL Reference Manual* for more information.

FRSY-023 WARN No blocks were purged

Cause: No blocks were purged for one of the following reasons: 1) No garbage blocks exist. 2) No spare blocks exist because the FROM disk is full.
Remedy: If you require more blocks, you must backup all your files, reformat the device, and restore the files. Refer to chapter 9, "File System", in the *FANUC Robotics SYSTEM R-J2 KAREL Reference Manual* for more information.

FRSY-024 WARN Purge is disabled

Cause: You are not allowed to purge the FROM disk because purge is disabled.
Remedy: Set \$PURGE_ENBL to TRUE and retry the purge operation. You may wish to set \$PURGE_ENBL to FALSE before running a program or application which requires fast cycle time.

FRSY-026 WARN CRC check failed

Cause: One or more files on the FROM disk are corrupted. This may occur if the FROM is wearing out.
Remedy: You should backup all your files, reformat the device, and restore the files. Refer to chapter 9, "File System", in the *FANUC Robotics SYSTEM R-J2 KAREL Reference Manual* for more information. If the problem persists, the FROM may need to be replaced.

FRSY-028 WARN %d out of %d bad FROM blocks

Cause: The FROM disk is wearing out.
Remedy: The system will continue to operate as long as enough blocks are available. When too many blocks become bad, the FROM will need to be replaced.

HOST Error Codes (ID = 67)

HOST-001 WARN End of directory reached

Cause: Your listing has reached the end of the directory. You do not have to do anything for this warning message.

Remedy: This is a notification.

HOST-002 WARN File already exists

Cause: The file name you are trying to create already exists on this device.

Remedy: Delete the file on this device or choose a different file name.

HOST-003 WARN File does not exist

Cause: The file you are trying to open does not exist on this device.

Remedy: Open a file that exists on the device.

HOST-004 WARN Illegal command received

Cause: The requested operation is not supported.

Remedy: Use only supported operations, or check command syntax.

HOST-005 WARN Disk is full

Cause: The disk file capacity has been reached.

Remedy: Delete some unneeded files or use a disk with sufficient free space.

HOST-006 WARN End of file reached

Cause: The end of the file was reached while reading.

Remedy: Do not attempt to read beyond the end of a file.

HOST-008 WARN Only one file may be opened

Cause: An attempt was made to open more than one file.

Remedy: Do not attempt to open more than one file at a time.

HOST-100 WARN Communications error

Cause: The protocol format was invalid.

Remedy: Retry the operation.

HOST-101 WARN Directory read error

Cause: The directory information is corrupted and unreadable.

Remedy: Try another disk or reformat the disk.

HOST-102 WARN Block check error

Cause: The checksum data is bad. Data is corrupted on the disk and can not be read.

Remedy: Try another disk, or reformat the disk

HOST-103 WARN Seek error

Cause: There is a bad sector or track on the disk.

Remedy: Clean the disk drive, try another disk, or reformat the disk.

HOST-104 WARN Disk timeout

Cause: The drive did not respond to a command.

Remedy: Check the cable to the drive and make sure drive power is on.

HOST-105 WARN Write protection violation

Cause: The disk has write protection enabled.

Remedy: Remove write protection from the disk or use a disk that is not write protected.

HOST-106 WARN \$PROTOENT entry not found

Cause: Protocol Entry structure (\$PROTOENT) is invalid. It should be reset to default values.

Remedy: Return Protocol Entry structure to initial values from Setup and Operations manual.

HOST-107 WARN \$SERVENT entry not found

Cause: Server Entry structure (\$SERVENT) is invalid. It should be reset to default values.

Remedy: Return Server Entry structure to initial values from Setup and Operations manual.

HOST-108 WARN Internet address not found

Cause: Internet Address needs to be set.

Remedy: Set Internet Address in the Host Comm TCP/IP Protocol Setup Menu.

HOST-109 WARN Host name not found

Cause: Host Name needs to be set.

Remedy: Set Host Name and Internet Address in The Host Comm TCP/IP Protocol Setup Menu.

HOST-110 WARN Node not found

Cause: The Remote Node Name needs to be set.
Remedy: Set Remote Node Name in the Host Comm TCP/IP Protocol Setup Menu.

HOST-111 WARN Cycle power to use Ethernet

Cause: ER-1 or ER-2 hardware is already running and can not be restarted without cycling power.
Remedy: Turn off and then turn on the controller.

HOST-126 WARN Invalid Ethernet address

Cause: The Ethernet address needs to be set.
Remedy: Set the Ethernet address in BMON.

HOST-127 WARN Ethernet firmware not loaded

Cause: The Ethernet Board firmware is not loaded.
Remedy: Load the Ethernet Board firmware in BMON.

HOST-128 WARN Ethernet hardware not installed

Cause: The Ethernet Board needs to be reinitialized.
Remedy: Install the Ethernet Board.

HOST-129 WARN Receiver error

Cause: Data received from external device is invalid. Most likely caused by electrical noise on receivers.
Remedy: The error can be cleared by Stopping and Starting the Tag.

HOST-130 WARN Buffer alignment wrong

Cause: A buffer was passed to the Serial Port Driver which can not be accessed.
Remedy: Ensure program can run on this version of controller. You might need to retranslate your program.

HOST-131 WARN Wrong state

Cause: The Host Comm system can not execute the requested command in the present operating mode.
Remedy: Stop and Start the Host Comm Tag to reset the operating mode.

HOST-132 WARN Can't allocate memory

Cause: The Host Comm system can not allocate memory buffers for receiving or transmitting messages
Remedy: Either add more memory to the controller or reduce the number of simultaneous connections.

HOST-133 WARN Wrong setup conditions

Cause: The Host Comm system is receiving messages but can not decode them.
Remedy: Correct port settings: data rate, data size, stop bits, etc to match external device.

HOST-134 WARN BCC or CRC error

Cause: The Host Comm system is receiving checksum errors on all messages.
Remedy: Ensure that the external device is using the same protocol.

HOST-135 WARN Timeout

Cause: There has not been any network activity on the Comm Tag for a period specified by Inactivity Timeout.
Remedy: Restart the Comm Tag.

HOST-136 WARN Device not ready

Cause: The remote device is connected but is not responding to requests.
Remedy: Check cabling between the devices and/or insure the device is powered.

HOST-137 WARN Request cancelled

Cause: The remote device indicates the operation was successfully terminated.
Remedy: The cancel command was successful.

HOST-138 WARN Request aborted

Cause: The remote device did not indicate operation was terminated.
Remedy: The command might have been completed before the cancel command was received.

HOST-139 WARN Invalid function

Cause: The Host Comm Protocol does not support the requested function.
Remedy: Check the Host Comm Protocol version to ensure the function is supported.

HOST-140 WARN Device offline

Cause: The remote device is connected but it is not online.
Remedy: Set the remote device online.

HOST-141 WARN Mount/Dismount error

Cause: The Host Comm Protocol could not be started on the selected Comm Tag.
Remedy: Either use another Comm Tag or Stop and Undefine the selected Comm Tag.

HOST-142 WARN Connection error

Cause: The Host Comm Protocol could not establish communication with the remote device. Possible software mismatch.
Remedy: Ensure both local and remote are using compatible software versions.

HOST-143 WARN Packet returned by close

Cause: The selected hardware port defined for the Comm Tag could not be closed.
Remedy: Power the controller off and then on and try again. If the error occurs again a cabling or hardware problem might exist with the port.

HOST-144 WARN No such device or address

Cause: The Comm Tag either does not have a protocol defined or if required does not have a port assigned.
Remedy: DEFINE a protocol to the Comm Tag or assign a port.

HOST-145 WARN Permission denied

Cause: An attempt has been made either to read a file opened for write access only or to write a file opened for read access only.
Remedy: Close and reopen the file with the correct access parameters.

HOST-146 WARN Bad address

Cause: A bad address has been detected.
Remedy: UNDEFINE and then DEFINE the Comm Tag after checking whether the Tag has a supported protocol.

HOST-147 WARN Block device required

Cause: The selected protocol requires a device port.
Remedy: First ensure the Port has No Use from Port Init Setup. Then assign it to the selected Comm Tag.

HOST-148 WARN Mount device busy

Cause: Either the Comm Tag is STARTED or it is presently in use.
Remedy: Either STOP the Comm Tag or select another Tag.

HOST-149 WARN No such device

Cause: The passed Device Type is not a Comm Tag type (Cx or Sx).
Remedy: Only Comm Tags can be used with this command.

HOST-150 WARN Invalid argument

Cause: The system does not support selected protocol.
Remedy: Either select another protocol or install the selected protocol.

HOST-151 WARN No more Ethernet buffers.

Cause: The System has run out of buffers to communicate with the Ethernet Remote PCB.
Remedy: Reduce the number of simultaneous connections as there is not enough memory.

HOST-152 WARN MAP: MIB not responding

Cause: If the MAP Interface Board is installed it is no longer responding to the MAIN CPU PCB.
Remedy: Replace the MAP Interface Board.

HOST-153 WARN MAP: PDU size too big

Cause: Either the received or transmitted Protocol Data Unit (PDU) is too big to fit in the buffer sizes which MAP is using.
Remedy: Increase size of PDU buffers by increasing the Host PDU Size (\$HOST_PDUSIZ), see Setup and Operations manual for maximum.

HOST-154 WARN MAP: Directory file missing

Cause: Directory file (umap_2_d.tx) is missing on RAM Disk
Remedy: Directory file contains node names and addresses. Load a saved version or recreate from distribution disks.

HOST-155 WARN MAP: Network file missing

Cause: Network file (umap_2_p.tx) is missing on the RAM Disk.
Remedy: Network file contains Station Address, slot time, and so forth. Load a saved version or recreate from distribution disks.

HOST-156 WARN MAP: invalid Local Appl. Name

Cause: Local Name is name of robot node. Host Name (\$HOST_NAME) must match the local Directory entry. It is missing in Directory File (umap_2_d.tx).

Remedy: Add Host Name as Local Name to Directory File or add Directory File local entry to Host Name (\$HOST_NAME) via MAP Protocol Setup Menu.

HOST-157 WARN MAP: invalid Remote Appl. Name

Cause: Can not find Remote Name in Directory File.

Remedy: Add Remote Name to Directory File via MAP Protocol Setup Menu.

HOST-158 WARN FTP: no connection available

Cause: An error occurred in the networking software.

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HOST-159 WARN FTP: login failed

Cause: The Comm Tag does not have a valid username and password.

Remedy: Enter a valid username and password for the Comm Tag.

HOST-160 WARN FTP: dismount request ignored

Cause: An error occurred in the networking software.

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HOST-161 WARN FTP: need remote host name

Cause: The Comm Tag does not have a remote host defined.

Remedy: Enter a remote host name to Current Remote and Startup Remote.

HRTL Error Codes (ID = 66)

HRTL-002 WARN File/Comm Tag does not exist

Cause: Either the file or the Comm Tag could not be found.

Remedy: Either retype the file name or DEFINE the Comm Tag.

HRTL-003 WARN No such process

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-004 WARN Interrupted system call

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-006 WARN No protocol or device dest

Cause: The Comm Tag either does not have a protocol defined or if required does not have a port assigned.

Remedy: DEFINE a protocol to the Comm Tag or assign a port.

HRTL-009 WARN Bad file number

Cause: The file number passed does not match with any open files.

Remedy: Copy the conditions which caused this to occur.

HRTL-013 WARN Access permission denied

Cause: An attempt has been made to either read a file opened for write access only or write a file open for read access only.

Remedy: Close and reopen the file with the correct access parameters.

HRTL-014 WARN Invalid Comm Tag

Cause: A bad address has been detected.

Remedy: UNDEFINE and then DEFINE the Comm Tag after checking the Tag has a supported protocol.

HRTL-015 WARN Port device required

Cause: The selected protocol requires a device port.

Remedy: First ensure the Port has No Use from Port Init Setup. Then assign it to the selected Comm Tag.

HRTL-016 WARN Comm Tag already defined

Cause: Either the Comm Tag is STARTED or it's presently in use.

Remedy: Either STOP the Comm Tag or select another Tag.

HRTL-017 WARN File exists

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-019 WARN Invalid device type

Cause: The passed Device Type is not a Comm Tag type (Cx or Sx).

Remedy: Only Comm Tags can be used with this command.

HRTL-022 WARN Invalid argument

Cause: The passed Device Type is not a Comm Tag type (Cx or Sx).

Remedy: Only Comm Tags can be used with this command.

HRTL-032 WARN Broken pipe

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-035 WARN Operation would block

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-036 WARN Operation now in progress

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-037 WARN Operation already in progress

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-039 WARN Destination address required

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-040 WARN Message too long

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-041 WARN Protocol wrong type

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-042 WARN Protocol not available

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-043 WARN Protocol not supported

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-045 WARN Operation not supported

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-047 WARN Address family not supported

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-048 WARN Address already in use

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-049 WARN Can't assign requested address

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-050 WARN Network is down

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-051 WARN Network is unreachable

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-053 WARN Software connection abort

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-054 WARN Connection reset by peer

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-055 WARN No buffer space available

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-056 WARN Socket is already connected

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-057 WARN Socket is not connected

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-060 WARN Connection timed out

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-061 WARN Connection refused

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-064 WARN Host is down

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

HRTL-065 WARN No route to host

Cause: An error occurred in the Ethernet networking software (TCP/IP).

Remedy: Consult your network administrator. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

INTP Error Codes (ID = 12)

- INTP-001 PAUSE.G Cannot lock the motion grp**
Cause: Motion control for the specified group cannot be locked.
Remedy: Check the teach pendant enable switch and other running programs to determine who has motion control.
- INTP-002 ABORT.G Program manager internal error**
Cause: Internal system error.
Remedy: Please contact FANUC Robotics hot line.
- INTP-003 ABORT.G Invalid request**
Cause: Internal system error.
Remedy: Please contact FANUC Robotics hot line.
- INTP-004 PAUSE.G Cannot ATTACH with TP enabled**
Cause: The ATTACH statement requires the teach pendant to be disabled.
Remedy: Disable the teach pendant.
- INTP-005 PAUSE.G Cannot release motion control**
Cause: Motion control cannot be released.
Remedy: Abort the running or paused program.
- INTP-100 ABORT.L (%s^4, %d^5) Internal error (PXnn)**
Cause: Internal system error.
Remedy: Contact FANUC Robotics hot line.
- INTP-101 ABORT.L (%s^4, %d^5) Internal error (system)**
Cause: Internal system error.
Remedy: Contact FANUC Robotics hot line.
- INTP-102 ABORT.L (%s^4, %d^5) Code format is invalid**
Cause: Program data is corrupted.
Remedy: For TPE programs, if possible, reload program from back-up device. If a back-up is not available, it may be necessary to re-create the particular routine. For KAREL programs, re-translate and re-load the program.
- INTP-103 ABORT.L (%s^4, %d^5) Program error**
Cause: An error occurred while the program was running.
Remedy: Refer to the error cause code.
- INTP-104 ABORT.L (%s^4, %d^5) Single step failed**
Cause: Single step cannot be executed
Remedy: Refer to the error cause code.
- INTP-105 ABORT.L (%s^4, %d^5) Run request failed**
Cause: Program cannot be started.
Remedy: Refer to the error cause code.
- INTP-106 PAUSE.L (%s^4, %d^5) Continue request failed**
Cause: Program cannot be resumed.
Remedy: Refer to the error cause code.
- INTP-107 ABORT.L (%s^4, %d^5) Pause request failed**
Cause: An error occurred when program execution was held.
Remedy: Refer to the error cause code.
- INTP-108 ABORT.L (%s^4, %d^5) Abort request failed**
Cause: An error occurred when program execution was aborted.
Remedy: Refer to the error cause code.
- INTP-109 WARN (%s^4, %d^5) BWD motion request failed**
Cause: Backward motion cannot be executed.
Remedy: Refer to the error cause code.
- INTP-110 WARN (%s^4, %d^5) Get task status request failed**
Cause: The specified task attribute is not found or is not read accessible.
Remedy: Check the attribute.
- INTP-111 WARN (%s^4, %d^5) Skip statement request failed**
Cause: The currently executing line cannot be changed.
Remedy: Refer to the error cause code.

INTP-112 PAUSE.L Cannot call interrupt routine

Cause: When this error code is issued with "MEMO-004 Specified program is in use", the action program is editing, pausing or executing.

Remedy: Please select the another program by select menu.

Cause: When this error code is issued with "PROG-020 Task is already aborted", the action program can not execute because the program which start the monitor is already aborted.

Remedy: In the program monitor, the action program can execute during the program which start the monitor is running.

INTP-113 PAUSE.L (%s^4, %d^5) Stop motion request failed

Cause: An error occurred when motion was stopped.

Remedy: Refer to the error cause code.

INTP-114 PAUSE.L (%s^4, %d^5) Cancel motion request failed

Cause: An error occurred when motion was canceled.

Remedy: Refer to the error cause code.

INTP-115 PAUSE.L (%s^4, %d^5) Resume motion request failed

Cause: An error occurred when motion was resumed.

Remedy: Refer to the error cause code.

INTP-116 PAUSE.L (%s^4, %d^5) Hold motion request failed

Cause: An error occurred when motion was held.

Remedy: Refer to the error cause code.

INTP-117 PAUSE.L (%s^4, %d^5) Unhold motion request failed

Cause: An error occurred when motion was unheld.

Remedy: Refer to the error cause code.

INTP-118 PAUSE.L (%s^4, %d^5) Walk back data request failed

Cause: An error occurred trying to obtain the execution history.

Remedy: Refer to the error cause code.

INTP-119 PAUSE.L (%s^4, %d^5) Get trace data request failed

Cause: An error occurred trying to obtain the trace data.

Remedy: Refer to the error cause code.

INTP-120 PAUSE.L (%s^4, %d^5) Unwait action request failed

Cause: An error occurred trying to continue program execution.

Remedy: Refer to the error cause code.

INTP-121 PAUSE.L (%s^4, %d^5) Release inquiry request failed

Cause: An error occurred trying to obtain motion information for the RELEASE statement.

Remedy: Refer to the error cause code.

INTP-122 PAUSE.L (%s^4, %d^5) Process motion data failed

Cause: An error occurred during process motion.

Remedy: Refer to the error cause code.

INTP-123 PAUSE.L (%s^4, %d^5) Process application data failed

Cause: An error occurred during process application.

Remedy: Refer to the error cause code.

INTP-124 ABORT.L (%s^4, %d^5) Invalid ITR routine

Cause: The specified interrupt routine is not a valid type.

Remedy: Refer to the error cause code.

INTP-125 ABORT.L Failed to convert position

Cause: The conversion of one position type to another failed.

Remedy: Refer to the error cause code.

INTP-126 ABORT.L Vision built-in return failed

Cause: The vision built-in failed to return.

Remedy: Refer to the error cause code.

INTP-127 WARN Power fail detected

Cause: Power failure was detected.

Remedy: Resume the program after hot start is complete.

INTP-128 PAUSE.L Pos reg is locked**Cause:** Position register is locked.**Remedy:** Wait a moment and try accessing the position register again.**INTP-129 ABORT.L Cannot use motion group****Cause:** Try to lock motion group even though this program cannot use motion group**Remedy:** Clear motion group mask in program detail screen**INTP-130 ABORT.L (%s^4, %d^5) Exec status recovery failed****Cause:** Failed to recover execution status.**Remedy:** Refer to the error cause code.**INTP-131 ABORT.L Number of stop exceeds limit****Cause:** Too many stop data is created at one time.**Remedy:** Decrease number of stop data.**INTP-200 PAUSE.L (%s^4, %d^5) Unimplemented TP instruction****Cause:** The teach pendant program instruction is not available.**Remedy:** Check the appropriate option is loaded.**INTP-201 PAUSE.L (%s^4, %d^5) Untaught element encountered****Cause:** The instruction is not taught.**Remedy:** Teach the instruction.**INTP-202 PAUSE.L (%s^4, %d^5) Syntax error****Cause:** Instruction syntax error.**Remedy:** Reteach the instruction.**INTP-203 PAUSE.L (%s^4, %d^5) Variable type mismatch****Cause:** The variable type is not correct.**Remedy:** Check the variable type.**INTP-204 PAUSE.L (%s^4, %d^5) Invalid value for index****Cause:** The index value is invalid.**Remedy:** Check the index value.**INTP-205 PAUSE.L (%s^4, %d^5) Analog port access error****Cause:** Analog I/O is not functioning properly.**Remedy:** Refer to the error cause code.**INTP-206 PAUSE.L (%s^4, %d^5) Digital port access error****Cause:** Digital I/O is not functioning properly.**Remedy:** Refer to the error cause code.**INTP-207 PAUSE.L (%s^4, %d^5) Group I/O port access error****Cause:** Group I/O is not functioning properly.**Remedy:** Refer to the error cause code.**INTP-208 PAUSE.L (%s^4, %d^5) Divide by 0****Cause:** Division by 0 was executed.**Remedy:** Check the value.**INTP-209 PAUSE.L (%s^4, %d^5) SELECT is needed****Cause:** A CASE instruction was executed before a SELECT instruction.**Remedy:** Add a SELECT instruction before the CASE instruction.**INTP-210 PAUSE.L (%s^4, %d^5) Start TIMER failed****Cause:** The program timer cannot be started.**Remedy:** Refer to the error cause code.**INTP-211 PAUSE.L (%s^4, %d^5) Delete TIMER failed****Cause:** The program timer cannot be stopped.**Remedy:** Refer to the error cause code.**INTP-212 PAUSE.L (%s^4, %d^5) Invalid value for OVERRIDE****Cause:** The indicated value cannot be used for the OVERRIDE instruction.**Remedy:** Check the value.

-
- INTP-213 PAUSE.L (%s^7 (%s^4, %d^5) UALM[%d^9]**
Cause: A user alarm occurred.
Remedy: Refer to the user alarm code.
- INTP-214 PAUSE.L (%s^4, %d^5) Specified group not locked**
Cause: The position register or frame setup instructions were executed in a program without a motion group.
Remedy: Set up the motion group in the program DETAIL screen.
- INTP-215 PAUSE.L (%s^4, %d^5) Group mismatch**
Cause: The position data is invalid.
Remedy: Check the position data.
- INTP-216 PAUSE.L (%s^4, %d^5) Invalid value for group number**
Cause: The indicated value is invalid for the motion group number.
Remedy: Check the value.
- INTP-217 PAUSE.L (%s^4, %d^5) SKIP CONDITION needed**
Cause: The SKIP instruction was executed before a SKIP CONDITION instruction.
Remedy: Add a SKIP CONDITION instruction.
- INTP-218 PAUSE.L (%s^4, %d^5) Skip failed**
Cause: The SKIP instruction or SKIP CONDITION instruction cannot be executed.
Remedy: Refer to the error cause code.
- INTP-219 ABORT.L (%s^4, %d^5) Pause task failed**
Cause: The PAUSE instruction cannot be executed.
Remedy: Refer to the error cause code.
- INTP-220 ABORT.L (%s^4, %d^5) Abort task failed**
Cause: The ABORT instruction cannot be executed.
Remedy: Refer to the error cause code.
- INTP-221 PAUSE.L (%s^4, %d^5) Application failed**
Cause: The application instruction cannot be executed.
Remedy: Refer to the error cause code.
- INTP-222 PAUSE.L (%s^4, %d^5) Call program failed**
Cause: The program CALL instruction cannot be executed.
Remedy: Refer to the error cause code.
- INTP-223 PAUSE.L (%s^4, %d^5) Delay time failed**
Cause: The WAIT instruction cannot be executed.
Remedy: Refer to the error cause code.
- INTP-224 PAUSE.L (%s^4, %d^5) Jump label failed**
Cause: The BRANCH instruction cannot be executed.
Remedy: Refer to the error cause code.
- INTP-225 PAUSE.L (%s^4, %d^5) Motion statement failed**
Cause: The MOTION instruction cannot be executed.
Remedy: Refer to the error cause code.
- INTP-226 PAUSE.L (%s^4, %d^5) Read position register failed**
Cause: The position register cannot be read.
Remedy: Refer to the error cause code.
- INTP-227 PAUSE.L (%s^4, %d^5) Write position register failed**
Cause: The position register cannot be written.
Remedy: Refer to the error cause code.
- INTP-228 PAUSE.L (%s^4, %d^5) Read register failed**
Cause: The register cannot be read.
Remedy: Refer to the error cause code.
- INTP-229 PAUSE.L (%s^4, %d^5) Write register failed**
Cause: The register cannot be written.
Remedy: Refer to the error cause code.

- INTP-230 PAUSE.L (%s^4, %d^5) Wait condition failed**
Cause: A condition WAIT instruction cannot be executed.
Remedy: Refer to the error cause code.
- INTP-231 PAUSE.L (%s^4, %d^5) Read next line failed**
Cause: The next line cannot be read.
Remedy: Refer to the error cause code.
- INTP-232 PAUSE.L (%s^4, %d^5) Invalid frame number**
Cause: The frame number is invalid.
Remedy: Check the frame number.
- INTP-233 PAUSE.L (%s^4, %d^5) Read frame value failed**
Cause: The specified frame cannot be read.
Remedy: Refer to the error cause code.
- INTP-234 PAUSE.L (%s^4, %d^5) Write frame value failed**
Cause: The specified frame cannot be written.
Remedy: Refer to the error cause code.
- INTP-235 PAUSE.L (%s^4, %d^5) Read pos item failed**
Cause: The position variable cannot be read.
Remedy: Refer to the error cause code.
- INTP-236 PAUSE.L (%s^4, %d^5) Write pos item failed**
Cause: The position variable cannot be written.
Remedy: Refer to the error cause code.
- INTP-237 WARN (%s^4, %d^5) No more motion for BWD**
Cause: Backward execution cannot be executed any more because the current program line is at the top.
Remedy: Do not use backward execution at this point
- INTP-238 WARN (%s^4, %d^5) BWD execution completed**
Cause: Backward execution was completed.
Remedy: Do not use backward execution from this point.
- INTP-239 WARN (%s^4, %d^5) Cannot execute backwards**
Cause: This instruction cannot be executed backwards.
Remedy: Set the cursor to the following line.
- INTP-240 PAUSE.L (%s^4, %d^5) Incompatible data type**
Cause: The specified data type in the PARAMETER instruction is invalid for the parameter type.
Remedy: Check the data type.
- INTP-241 PAUSE.L (%s^4, %d^5) Unsupported parameter**
Cause: This type of parameter cannot be used.
Remedy: Check the parameter type.
- INTP-242 PAUSE.L (%s^4, %d^5) Offset value is needed**
Cause: An OFFSET instruction was executed before an OFFSET CONDITION instruction. A position register was not taught in the OFFSET PR[] instruction.
Remedy: Add an OFFSET CONDITION instruction before the OFFSET instruction. Teach the position register.
- INTP-243 ABORT.L (%s^4, %d^5) Def grp is not specified**
Cause: This program has no motion group defined. The MOTION instruction cannot be executed.
Remedy: Remove the MOTION instruction or set up the motion group in the program DETAIL screen.
- INTP-244 PAUSE.L (%s^4, %d^5) Invalid line number**
Cause: The input line number is incorrect.
Remedy: Check the line number.
- INTP-245 PAUSE.L (%s^4, %d^5) RCV stmt failed**
Cause: The RECEIVE R[] instruction cannot be executed.
Remedy: Refer to the error cause code.
- INTP-246 PAUSE.L (%s^4, %d^5) SEMAPHORE stmt failed**
Cause: The SEMAPHORE instruction cannot be executed.
Remedy: Refer to the error cause code.

INTP-247 PAUSE.L (%s^4, %d^5) Pre exec failed

Cause: Pre-planned execution of the program failed.

INTP-248 PAUSE.L (%s^4, %d^5) MACRO failed

Cause: The MACRO instruction cannot be executed.

Remedy: Refer to the error cause code.

INTP-249 PAUSE.L Macro is not set correctly

Cause: The MACRO setup was invalid.

Remedy: Check the MACRO setup.

INTP-250 PAUSE.L (%s^4, %d^5) Invalid uframe number

Cause: The user frame number is invalid.

Remedy: Refer to the error cause code.

INTP-251 PAUSE.L (%s^4, %d^5) Invalid utool number

Cause: The tool frame number is invalid.

Remedy: Refer to the error cause code.

INTP-252 PAUSE.L User frame number mismatch

Cause: The user frame number in the positional data is not the same as the currently selected user frame number.

Remedy: Check the user frame number.

INTP-253 PAUSE.L Tool frame number mismatch

Cause: The tool frame number in the positional data is not the same as the currently selected tool frame number.

Remedy: Check the tool frame number.

INTP-254 PAUSE.L (%s^4, %d^5) Parameter not found

Cause: The specified parameter name cannot be found.

Remedy: Check the parameter name.

INTP-255 PAUSE.L (%s^4, %d^5) CAL_MATRIX failed

Cause: The CAL_MATRIX instruction cannot be executed.

Remedy: Refer to the error cause code.

INTP-256 PAUSE.L (%s^4, %d^5) No data for CAL_MATRIX

Cause: The origin 3 points or destination 3 points are not taught.

Remedy: Teach the origin 3 points or destination 3 points.

INTP-257 PAUSE.L (%s^4, %d^5) Invalid delay time

Cause: The wait time value is negative or exceeds the maximum value of 2147483.647 sec.

Remedy: Input a correct value.

INTP-258 PAUSE.L (%s^4, %d^5) Weld port access error

Cause: The weld is not functioning properly.

Remedy: Refer to the error cause code.

INTP-259 PAUSE.L (%s^4, %d^5) Invalid position type

Cause: The data type of the position register was taught using joint type.

Remedy: Change position register data to Cartesian.

INTP-260 PAUSE.L (%s^4, %d^5) Invalid torque limit value

Cause: Invalid torque value.

Remedy: Input a correct value.

INTP-261 PAUSE.L (%s^4, %d^5) Array subscript missing

Cause: A subscript is missing from a TPE PARAMETER statement that specifies an array.

Remedy: Correct the PARAMETER statement to include the subscript of the desired array element.

INTP-262 PAUSE.L (%s^4, %d^5) Field name missing

Cause: A field name is required in a PARAMETER statement that specifies a structure.

Remedy: Correct the PARAMETER statement to include the name of the desired field

INTP-263 PAUSE.L (%s^4, %d^5) Invalid register type

Cause: The register type is not valid.

Remedy: Check the register type.

- INTP-264 AGSVOF (%s^4, %d^5) Soft float time out**
- INTP-265 PAUSE.L (%s^4, %d^5) Invalid value for speed value**
Cause: The indicated value cannot be used for the AF instruction.
Remedy: Check the value.
- INTP-266 ABORT.L (%s^4, %d^5) Mnemonic in interrupt is failed**
Cause: There isn't CANCEL or STOP instruction.
Remedy: insert CANCEL or STOP before call interrupt routine.
- INTP-267 PAUSE.L (%s^4, %d^5) RUN stmt failed**
Cause: Specified program is already running
Remedy: Abort specified program
- INTP-268 PAUSE.L (%s^4, %d^5) This statement only one in each line**
Cause: This statement can exist in one in each line
Remedy: delete statement
- INTP-269 PAUSE.L (%s^4, %d^5) Skip statement only one in each line**
Cause: Skip statement can exist in one in each line
Remedy: delete skip statement
- INTP-270 PAUSE.L (%s^4, %d^5) different group cannot BWD**
Cause: During backward execution, a move is encountered that has a different group number from the previous motion statement.
Remedy: Use FWD carefully
- INTP-271 WARN (%s^4, %d^5) Excessive torque limit value**
Cause: Torque limit value exceeds maximum value. Torque limit value was modified to the maximum value.
Remedy: Set torque limit value less than or equal to the maximum value.
- INTP-272 PAUSE.L (%s^4, %d^5) Unsupported operator**
Cause: This operator is not supported.
Remedy: Check the operator
- INTP-274 PAUSE.L (%s^4, %d^5) CH program error**
Cause: The specified ch program has illegal instructions.
Remedy: Refer to the error cause code and modify the ch program correctly.
Cause: When this error code is issued with "MEMO-004 Specified program is in use", the specified ch program is editing, pausing or executing.
Remedy: Please select the another program by select menu.
- INTP-275 PAUSE Invalid sub type of CH program**
Cause: The sub type of specified ch program cannot be used.
Remedy: Check the sub type of this CH program.
- INTP-276 PAUSE (%s^4, %d^5) Invalid combination of motion option**
Cause: The motion option instructions (SKIP, TIME BEFORE/AFTER, and application instruction) cannot be taught together.
Remedy: Delete the motion option instruction.
- INTP-277 PAUSE (%s^4, %d^5) Internal MACRO EPT data mismatch**
Cause: The EPT index in macro table doesn't point the program name defined in macro table. That is, the EPT index in macro table is incorrect.
Remedy: Please set the correct EPT index for the program name defined in macro table.
- INTP-278 PAUSE %s^7**
Cause: The DI monitor alarm for auto error recovery function occurs.
Remedy: This alarm is defined by the customer. Therefore the customer knows the remedy for this alarm.
- INTP-279 PAUSE (%s^4, %d^5) Application data mismatch**
Cause: The application instruction was executed. But this application instruction doesn't match to the application process data of this program.
Remedy: Please change the application process data of this program to the adequate application for this application instruction.
- INTP-280 PAUSE (%s^4, %d^5) Application data mismatch**
Cause: The application data of called program is different from that of the original program.
Remedy: Please change the structure of program.

-
- INTP-281 PAUSE (%s^4, %d^5) No application data**
Cause: This program doesn't have the application data.
Remedy: Please define the application data in the program detail screen.
- INTP-283 PAUSE (%s^4, %d^5) Stack over flow for fast fault recovery**
Cause: Stack over flow to record the fast fault recovery nesting data.
Remedy: Reduce the nesting of the program.
- INTP-284 PAUSE No detection of fast fault recovery**
Cause: The point for the fast fault recover cannot detected.
Remedy:
- INTP-285 WARN Karel program cannot entry in fast fault recovery**
Cause: The fast entry cannot be performed in the karel program.
Remedy: Use TP program.
- INTP-286 WARN MAINT program isn't defined in fast fault recovery**
Cause: MAINT program is not defined in fast fault recovery.
Remedy: Define the MAINT program.
- INTP-287 PAUSE Fail to execute MAINT program**
Cause: It failed to execute MAINT program.
Remedy: Confirm the MAINT program name is correct or MAINT program exist in actual.
- INTP-289 PAUSE Can't save ffast point at program change**
Cause: When fast fault is enabled, the program was paused at the part of program change.
Remedy: Check whether the CONT termination exists at end of sub-program. If exist, please change it to FINE. This is the limitation of the fast fault recovery function.
- INTP-290 PAUSE Fast fault recovery position is not saved**
Cause: During fast fault recovery sequence, any alarm occurs. So the fast fault recovery position is not saved.
Remedy: This message is for information purposes only.
- INTP-300 ABORT.L (%s^4, %d^5) Unimplemented P-code**
Cause: KAREL program error. This KAREL statement cannot be executed.
Remedy: Check the KAREL translator software version.
- INTP-301 ABORT.L (%s^4, %d^5) Stack underflow**
Cause: KAREL program error. Execution entered into a FOR loop by the GOTO statement.
Remedy: A GOTO statement cannot be used to enter or exit a FOR loop. Check the label of the GOTO statement.
- INTP-302 ABORT.L (%s^4, %d^5) Stack overflow**
Cause: The program stack overflowed. Too many local variables were declared or too many routines were called.
Remedy: For KAREL programs, refer to the KAREL Reference Manual, Stack Usage and the %STACKSIZE Translator Directive.
- INTP-303 ABORT.L (%s^4, %d^5) Specified value exceeds limit**
Cause: KAREL program error. The specified value exceeds the maximum limit.
Remedy: Check the value.
- INTP-304 ABORT.L (%s^4, %d^5) Array length mismatch**
Cause: KAREL program error. The dimensions of the arrays are not the same.
Remedy: Check the dimensions of the arrays.
- INTP-305 ABORT.L (%s^4, %d^5) Error related condition handler**
Cause: KAREL program error. A condition handler error occurred.
Remedy: Refer to the error cause code.
- INTP-306 ABORT.L (%s^4, %d^5) Attach request failed**
Cause: KAREL program error. The ATTACH statement failed.
Remedy: Refer to the error cause code.
- INTP-307 ABORT.L (%s^4, %d^5) Detach request failed**
Cause: KAREL program error. The DETACH statement failed.
Remedy: Refer to the error cause code.
- INTP-308 ABORT.L (%s^4, %d^5) No case match is encountered**
Cause: KAREL program error. The CASE statement does not match any branches.
Remedy: Check the CASE value and branches.

-
- INTP-309 ABORT.L (%s^4, %d^5) Undefined WITHCH parameter**
Cause: KAREL program error. The specified parameter cannot be used in the with clause of the condition handler.
Remedy: Check the parameter.
- INTP-310 ABORT.L (%s^4, %d^5) Invalid subscript for array**
Cause: KAREL program error. The index of the array is invalid.
Remedy: Check the length of the array and index value.
- INTP-311 PAUSE.L (%s^4, %d^5) Uninitialized data is used**
Cause: KAREL program error. Untaught or uninitialized data was used.
Remedy: Teach or initialize the data before using it.
- INTP-312 ABORT.L (%s^4, %d^5) Invalid joint number**
Cause: KAREL program error. The wrong axis number was used.
Remedy: Check the axis number and the data value.
- INTP-313 ABORT.L (%s^4, %d^5) Motion statement failed**
Cause: KAREL program error. The MOTION statement cannot be executed.
Remedy: Refer to the error cause code.
- INTP-314 ABORT.L (%s^4, %d^5) Return program failed**
Cause: KAREL program error. Execution cannot be returned from the routine.
Remedy: Refer to the error cause code.
- INTP-315 ABORT.L (%s^4, %d^5) Built-in execution failed**
Cause: KAREL program error. A built-in routine error occurred
Remedy: Refer to the error cause code.
- INTP-316 ABORT.L (%s^4, %d^5) Call program failed**
Cause: KAREL program error. The routine cannot be called.
Remedy: Refer to the error cause code. Verify the routine is loaded.
- INTP-317 ABORT.L (%s^4, %d^5) Invalid condition specified**
Cause: KAREL program error. The specified condition was invalid.
Remedy: Check the condition.
- INTP-318 ABORT.L (%s^4, %d^5) Invalid action specified**
Cause: KAREL program error. The specified action was invalid.
Remedy: Check the action.
- INTP-319 ABORT.L (%s^4, %d^5) Invalid type code**
Cause: KAREL program error. The data type was invalid.
Remedy: Check the data type.
- INTP-320 ABORT.L (%s^4, %d^5) Undefined built-in**
Cause: KAREL program error. The built-in routine is not defined.
Remedy: Check the appropriate option is loaded.
- INTP-321 ABORT.L (%s^4, %d^5) END stmt of a func rtn**
Cause: KAREL program error. The END statement was executed in a function routine instead of a RETURN statement.
Remedy: Add a RETURN statement to the function routine.
- INTP-322 ABORT.L (%s^4, %d^5) Invalid arg val for builtin**
Cause: KAREL program error. The argument value of a built-in routine was wrong.
Remedy: Check the argument value.
- INTP-323 ABORT.L (%s^4, %d^5) Value overflow**
Cause: KAREL program error. The data value for the variable was too large.
Remedy: Check the variable's type and data value.
- INTP-324 ABORT.L (%s^4, %d^5) Invalid open mode string**
Cause: KAREL program error. The usage string in the OPEN FILE statement was invalid.
Remedy: Check the usage string in the OPEN FILE statement.
- INTP-325 ABORT.L (%s^4, %d^5) Invalid file string**
Cause: KAREL program error. The file string in the OPEN FILE statement was invalid.
Remedy: Check the file string. If no device is specified, the default device is used.

- INTP-326 ABORT.L (%s^4, %d^5) File var is already used**
Cause: KAREL program error. The FILE variable is already being used.
Remedy: Close the file before reusing the FILE variable or add a new FILE variable.
- INTP-327 ABORT.L (%s^4, %d^5) Open file failed**
Cause: KAREL program error. The file could not be opened.
Remedy: Refer to the error cause code.
- INTP-328 ABORT.L (%s^4, %d^5) File is not opened**
Cause: KAREL program error. The specified file was not opened before operation.
Remedy: Open the file before operation.
- INTP-329 ABORT.L (%s^4, %d^5) Write variable failed**
Cause: KAREL program error. The value cannot be written to the variable.
Remedy: Refer to the error cause code.
- INTP-330 ABORT.L (%s^4, %d^5) Write file failed**
Cause: KAREL program error. Writing to the file failed.
Remedy: Refer to the error cause code.
- INTP-331 ABORT.L (%s^4, %d^5) Read variable failed**
Cause: KAREL program error. Reading the variable failed.
Remedy: Refer to the error cause code.
- INTP-332 ABORT.L (%s^4, %d^5) Read data is too short**
Cause: KAREL program error. Data read from the file is too short.
Remedy: Check the data in the file.
- INTP-333 ABORT.L (%s^4, %d^5) Invalid ASCII string for read**
Cause: KAREL program error. The string read from the file is wrong.
Remedy: Check the data of the file.
- INTP-334 ABORT.L (%s^4, %d^5) Read file failed**
Cause: KAREL program error. Reading from the file failed.
Remedy: Refer to the error cause code.
- INTP-335 ABORT.L (%s^4, %d^5) Cannot open pre-defined file**
Cause: KAREL program error. A file pre-defined by the system cannot be opened.
Remedy: Use the file defined by the system without opening it.
- INTP-336 ABORT.L (%s^4, %d^5) Cannot close pre-defined file**
Cause: KAREL program error. A file pre-defined by the system cannot be closed.
Remedy: Do not try to close it.
- INTP-337 ABORT.L (%s^4, %d^5) Invalid routine type**
Cause: KAREL program error. This routine cannot be used.
Remedy: Check the routine type and name.
- INTP-338 ABORT.L (%s^4, %d^5) Close file failed**
Cause: KAREL program error. Closing the file failed.
Remedy: Refer to the error cause code.
- INTP-339 ABORT.L (%s^4, %d^5) Invalid program name**
Cause: KAREL program error. The program name is invalid.
Remedy: Check the program name.
- INTP-340 ABORT.L (%s^4, %d^5) Invalid variable name**
Cause: KAREL program error. The variable name is invalid.
Remedy: Check the variable name.
- INTP-341 ABORT.L (%s^4, %d^5) Variable not found**
Cause: KAREL program error. The variable cannot be found.
Remedy: Verify the program name and variable name.
- INTP-342 ABORT.L (%s^4, %d^5) Incompatible variable**
Cause: KAREL program error. The data type defined by the BYNAME function and the variable type are mismatched.
Remedy: Check the data type and variable type.

-
- INTP-343 ABORT.L (%s^4, %d^5) Reference stack overflow**
Cause: KAREL program error. Too many variables are passed using the BYNAME function.
Remedy: Decrease the number of BYNAME functions.
- INTP-344 ABORT.L (%s^4, %d^5) Readahead buffer overflow**
Cause: KAREL program error. The buffer to read ahead from the device overflowed.
Remedy: Increase the buffer size.
- INTP-345 ABORT.L (%s^4, %d^5) Pause task failed**
Cause: KAREL program error. The PAUSE statement cannot be executed.
Remedy: Refer to the error cause code.
- INTP-346 ABORT.L (%s^4, %d^5) Abort task failed**
Cause: KAREL program error. The ABORT statement cannot be executed.
Remedy: Refer to the error cause code.
- INTP-347 ABORT.L (%s^4, %d^5) Read I/O value failed**
Cause: KAREL program error. The digital input signal cannot be input.
Remedy: Refer to the error cause code.
- INTP-348 ABORT.L (%s^4, %d^5) Write I/O value failed**
Cause: KAREL program error. The digital output signal cannot be output.
Remedy: Refer to the error cause code.
- INTP-349 ABORT.L (%s^4, %d^5) Hold motion failed**
Cause: KAREL program error. The HOLD statement cannot be executed.
Remedy: Refer to the error cause code.
- INTP-350 ABORT.L (%s^4, %d^5) Unhold motion failed**
Cause: KAREL program error. The UNHOLD statement cannot be executed.
Remedy: Refer to the error cause code.
- INTP-351 ABORT.L (%s^4, %d^5) Stop motion failed**
Cause: KAREL program error. The STOP statement cannot be executed.
Remedy: Refer to the error cause code.
- INTP-352 ABORT.L (%s^4, %d^5) Cancel motion failed**
Cause: KAREL program error. The CANCEL statement cannot be executed.
Remedy: Refer to the error cause code.
- INTP-353 ABORT.L (%s^4, %d^5) Resume motion failed**
Cause: KAREL program error. The RESUME statement cannot be executed.
Remedy: Refer to the error cause code.
- INTP-354 ABORT.L (%s^4, %d^5) Break point failed**
Cause: KAREL program error. The break point function cannot be executed.
Remedy: Refer to the error cause code.
- INTP-355 ABORT.L (%s^4, %d^5) AMR is not found**
Cause: KAREL program error. The AMR operated by the RETURN_AMR built-in routine was not found.
Remedy: Check program operation.
- INTP-356 ABORT.L (%s^4, %d^5) AMR is not processed yet**
Cause: KAREL program error. The RETURN_AMR built-in routine cannot be used for an unoperated AMR.
Remedy: Operate the AMR using the WAIT_AMR built-in routine.
- INTP-357 ABORT.L (%s^4, %d^5) WAIT_AMR is cancelled**
Cause: KAREL program error. The execution of the WAIT_AMR built-in routine was cancelled.
Remedy: The program executing the WAIT_AMR must be restarted.
- INTP-358 ABORT.L (%s^4, %d^5) Timeout at read request**
Cause: KAREL program error. The READ statement timed out.
Remedy: Check the device being read.
- INTP-359 ABORT.L (%s^4, %d^5) Read request is nested**
Cause: KAREL program error. Another READ statement was executed while a READ statement was waiting for input.
Remedy: Remove nested reads.

- INTP-360 ABORT.L (%s^4, %d^5) Vector is 0**
Cause: KAREL program error. The vector value was invalid.
Remedy: Check the vector value.
- INTP-361 PAUSE.L (%s^4, %d^5) FRAME:P2 is same as P1**
Cause: KAREL program error. The X-axis direction cannot be calculated in the FRAME built-in routine because P1 and P2 are the same point..
Remedy: Teach P1 and P2 as different points.
- INTP-362 PAUSE.L (%s^4, %d^5) FRAME:P3 is same as P1**
Cause: KAREL program error. The X-Y plane cannot be calculated in the FRAME built-in routine because P1 and P3 are the same point.
Remedy: Teach P1 and P3 as different points.
- INTP-363 PAUSE.L (%s^4, %d^5) FRAME:P3 exists on line P2-P1**
Cause: KAREL program error. The X-Y plane cannot be calculated in the FRAME built-in routine because P3 is located in the X-axis direction..
Remedy: Teach P3 out of the X-axis direction.
- INTP-364 ABORT.L (%s^4, %d^5) String too short for data**
Cause: KAREL program error. The target string was too short.
Remedy: Increase the target string size.
- INTP-365 ABORT.L (%s^4, %d^5) Predefined window not opened**
Cause: KAREL program error. A FILE pre-defined by the system is not opened.
Remedy: Check the use of this file.
- INTP-366 ABORT.L (%s^4, %d^5) I/O status is not cleared**
Cause: KAREL program error. The last file operation failed.
Remedy: Reset the error using the CLR_IO_STAT built-in routine.
- INTP-367 ABORT.L (%s^4, %d^5) Bad base in format**
Cause: KAREL program error. I/O mode operates only from binary to hexadecimal.
Remedy: Check the specified mode.
- INTP-368 PAUSE.L (%s^4, %d^5) Cannot use specified program**
Cause: KAREL program error. The specified program cannot be used.
Remedy: Refer to the error cause code.
- INTP-369 ABORT.L (%s^4, %d^5) Timeout at WAIT_AMR**
Cause: KAREL program error. The WAIT_AMR built-in routine timed out.
Remedy: If an AMR was expected within the time-out value check logic in the task that should have posted the AMR
- INTP-370 ABORT.L (%s^4, %d^5) Vision CPU not plugged in**
Cause: KAREL program error. The vision CPU board is not plugged in.
Remedy: Plug in the vision CPU board.
- INTP-371 ABORT.L (%s^4, %d^5) Vision built-in overflow**
Cause: KAREL program error. The operation overflowed in the vision built-in routine.
Remedy: Modify program so fewer vision builtins are executing at the same time.
- INTP-372 ABORT.L (%s^4, %d^5) Undefined vision built-in**
Cause: KAREL program error. The vision built-in routine is not defined.
Remedy: Check the appropriate option is loaded.
- INTP-373 ABORT.L (%s^4, %d^5) Undefined vision parameter type**
Cause: KAREL program error. The parameter to the vision built-in routine is invalid.
Remedy: Check the parameter of the vision built-in routine.
- INTP-374 ABORT.L (%s^4, %d^5) Undefined vision return type**
Cause: KAREL program error. The return value from the vision built-in routine is invalid.
Remedy: Check the return value from the vision built-in routine.
- INTP-375 ABORT.L (%s^4, %d^5) System var passed using BYNAME**
Cause: KAREL program error. System variables cannot be passed using the BYNAME function.
Remedy: Pass without using BYNAME or use GET_VAR and SET_VAR instead.

INTP-376 ABORT.L (%s^4, %d^5) Motion in interrupt is failed**Cause:** There isn't CANCEL or STOP instruction.**Remedy:** insert CANCEL or STOP before call interrupt routine.**INTP-377 WARN (%s^4, %d^5) Local COND recovery failed****Cause:** This local condition can't be recovered.**Remedy:** Refer to the error cause code.**INTP-378 WARN (%s^4, %d^5) Local variable is used****Cause:** Local variable or parameter is used for the condition.**Remedy:** Use global variable to recover local condition.**INTP-379 ABORT.L Bad condition handler number****Cause:** An invalid condition handler number was used in a condition handler definition, or an ENABLE, DISABLE, or PURGE statement or action**Remedy:** Correct the condition handler number. Condition handler numbers must be in the range 1-1000.**INTP-380 ABORT.L Bad program number****Cause:** A invalid program number has been specified in an ABORT PROGRAM, PAUSE PROGRAM, or CONTINUE PROGRAM condition or action.**Remedy:** Use a valid program number. Program numbers must be in the range 1..\$SCR,\$MAXNUMTASK + 2.**INTP-400 ABORT.L (%s^4, %d^5) Number of motions exceeded****Cause:** Too many motions are executed at the same time.**Remedy:** Execute the next motion after the completion of the last motion. Decrease the number of motions at the same time.**INTP-401 ABORT.L (%s^4, %d^5) Not On Top Of Stack****Cause:** Paused motion exists after the motion was resumed.**Remedy:** Resume the motion that was previously paused.

JOG Error Codes (ID = 19)

JOG-000 WARN Unknown error (MJ00)

Cause: System internal error
Remedy: Notify FANUC Robotics.

JOG-001 WARN Overtravel Violation

Cause: A robot overtravel has occurred
Remedy: Use the MANUAL FCTNS OT release menu in to find out which axis is in an overtravel condition. Release overtravel by holding the SHIFT key and pressing the RESET key. At this time the servo power will be turned on. If the SHIFT key is released, the servo power will be turned off again. You can only use JOINT to jog the axis out of overtravel. If you want to jog the overtraveled axis further into the overtravel direction, you have to release the axis by moving the cursor to the axis direction you want, then press release function key in the OT release menu. At this point you can jog the axis to that direction.

JOG-002 WARN Robot not Calibrated

Cause: Robot has not been calibrated
Remedy: Set the system variable \$MASTER_ENB to 1. This will cause the Master/Cal menu of SYSTEM menu to display. Select the calibrate item in this menu to calibrate the robot

JOG-003 WARN No Motion Control

Cause: Other program has motion control
Remedy: Abort the program that has motion control by pressing FCTN key then selecting ABORT.

JOG-004 WARN Illegal linear jogging

Cause: You cannot do more than one rotational jog at a time
Remedy: Only press one rotational jog key at a time.

JOG-005 WARN Can not clear hold flag

Cause: The system call to clear hold flag failed error.
Remedy: Perform a cycle start.

JOG-006 WARN Subgroup does not exist

Cause: No extended axis exist in this group with which to jog.
Remedy: None required

JOG-007 WARN Press shift key to jog

Cause: The SHIFT key is not pressed
Remedy: You must press the SHIFT key when jogging the robot. Release the jog key then hold the SHIFT key and press the jog key to jog.

JOG-008 WARN Turn on TP to jog

Cause: Teach pendant is not enable
Remedy: Hold the DEADMAN and turn on the teach pendant before jogging the robot.

JOG-009 WARN Hold deadman to jog

Cause: The DEADMAN switch is not pressed
Remedy: Press the DEADMAN switch, then press the RESET key to clear the error

JOG-010 WARN Jog pressed before shift

Cause: The jog key was pressed before the shift key was pressed
Remedy: Release the jog key. Then, hold the SHIFT key and press the jog key

JOG-011 WARN Utool changed while jogging

Cause: The selected tool frame changed while jogging
Remedy: Release the shift key and the jog key. The new TOOL frame will take effect automatically. To start jogging, press the shift and the jog key.

JOG-012 WARN manual brake enabled

Cause: The manual brake enabled
Remedy: Engage all the brakes by pressing EMERGENCY STOP button, then press the RESET key. To start jogging, press the shift and the jog key.

JOG-013 WARN Stroke limit (G:%d A:%x Hex)**Cause:** Robot axis reaches its specified stroke limit**JOG-014 WARN Vertical fixture position****Cause:** Robot reaches its vertical fixture position**JOG-015 WARN Horizontal fixture position****Cause:** Robot reaches its horizontal fixture position**Remedy:** None required**JOG-016 SERVO Softfloat time out(G:%d)****Cause:** Follow-up time is over when softfloat is ON**Remedy:** Make \$SFLT_FUPTIM larger.**JOG-020 PAUSE Can not PATH JOG****Cause:** PATH JOG has selected, but robot is not currently on a taught path, or tool Z direction is same teaching path, so Y direction can not be determined. Can not PATH JOG**Remedy:** Use shift-FWD to execute program path, or specify another jog frame.**JOG-021 PAUSE Multi key is pressed****Cause:** Use of multiple jog keys is not supported in PATH JOG**Remedy:** Use only one jog key at a time.

LANG Error Codes (ID = 21)**LANG-004 WARN File is not open**

Cause: (1) The wrong port is set to the port you want to use.
(2) The device might be out of order.

Remedy: (1) Set the correct port.
(2) Check the device if it works fine.

LANG-005 WARN Program type is different

Cause: It is not able to process except for a TPE program.

Remedy: Please select the TPE program.

LANG-014 WARN Program already exists

Cause: The program that is about to load, it has already existed in the system.

Remedy: If you may load it, please delete the program in the system.

LANG-015 WARN Can not write file

Cause: It failed that writing the data to the floppy.

Remedy: Please check the connection of the device.

LANG-016 WARN Can not read file

Cause: It failed that reading the data from the floppy.

Remedy: Please check the connection of the device.

LANG-017 WARN File format is incorrect

Cause: There was abnormality data when saving to a file. Or, the file data was broken.

Remedy: There is not means for avoid on the side of an user.

LANG-018 WARN Group mask value is incorrect

Cause: A group number of position data is duplicating.

Remedy: Please remake the position data that a group number does not duplicate.

LANG-094 WARN File already exists

Cause: The specified file already exists in the floppy.

Remedy: If you need to write the file, please delete the file in the floppy.

LANG-050 WARN WARN %s contains %s, program/file names must match

Cause: The file name and the program name is not same. Their names must match.

Remedy: Please rename the file name to be same as the program name.

LANG-095 WARN File does not exist

Cause: The specified file does not exist in the floppy.

Remedy: Please check the file name or content of the floppy.

LANG-096 WARN Disk is full

Cause: There is not the capacity of a floppy disk disk in the case that it is using a floppy disk.

Remedy: Please use a new floppy disk. Or, delete an necessary file in order to make the capacity for saving to the floppy.

LANG-098 WARN Disk timeout

Cause: It could not access the disk.

Remedy: Please check if the correct device is set to port and it turns on.

LANG-099 WARN Write protection violation

Cause: The disk has write protection.

Remedy: Please cancel the write protection.

LANG-100 WARN Device error

Cause: It could not access the device.

Remedy: Please connect the correct device to the correct port.

LNTK Error Codes (ID = 44)

LNTK-000 STOP Unknown error (LN00)

Cause: System internal error.

Remedy: Press RESET to clear the error and continue the program. If this error continues to occur, perform a cold start by turning off the robot, then while pressing SHIFT and RESET on the teach pendant, turn the robot back on. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

LNTK-001 STOP No global variables

Cause: The Line Track global variables have NOT been properly loaded.

Remedy: Check the application installation manual for the proper installation procedure for the Line Track system.

LNTK-002 STOP Motion data missing

Cause: The Line Track internal motion data was NOT found.

Remedy: Press RESET to clear the error and continue the program. If this error continues to occur, perform a cold start by turning off the robot, then while pressing SHIFT and RESET on the teach pendant, turn the robot back on. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

LNTK-003 STOP Error allocating data memory

Cause: The Line track internal memory allocation failed.

Remedy: Check Memory usage and Line Track installation.

LNTK-004 STOP No system variables

Cause: The Line Track system variables (eg. \$LNSCH[], \$LNSNRSCH[]) were not found.

Remedy: Check the application installation manual for the proper installation procedure for the Line Track system.

LNTK-005 STOP Illegal schedule number

Cause: An invalid Line Track schedule (track or frame) number was used within a program instruction (eg. TRK[59]) or program header data (eg. FRAME = 59).

Remedy: Check all schedule numbers (TRK[] or FRAME usages) used within the specified program to verify that they are within the allowable range specified for the \$LNSCH[] system variable.

LNTK-006 STOP Illegal tracking type

Cause: An invalid tracking type was specified within the tracking schedule (i.e. \$LNSCH[]) associated with the specified program.

Remedy: Check the value of \$LNSCH[i].\$TRK_TYPE (where 'i' is the FRAME number specified within the DETAIL screen for the specified program) to make sure that it is one of the valid values listed under the description for this system variable.

LNTK-007 STOP Illegal encoder number

Cause: An invalid sensor(encoder) number was used within the specified tracking program instruction or within the program's associated schedule \$LNSCH[i].\$TRK_ENC_NUM value (where 'i' is the FRAME number used within the DETAIL screen for the specified program).

Remedy: Check the value of the specified program instruction's LINE[] parameter and the program's associated schedule \$LNSCH[i].\$TRK_ENC_NUM value to make sure that it is one of the valid values listed under the description for the \$ENC_STAT[] system variable.

LNTK-008 STOP Invalid nominal position

Cause: An invalid or uninitialized nominal tracking frame position was used within the tracking schedule (i.e. \$LNSCH[]) associated with the specified program.

Remedy: Check the value of \$LNSCH[i].\$TRK_FRAME (where 'i' is the FRAME number specified within the DETAIL screen for the specified program) to make sure that it is a properly initialized, valid position.

LNTK-009 STOP Illegal position type

Cause: The position type used within the specified program is not valid.

Remedy: Check the KAREL or TPE user manual for valid position types.

LNTK-010 STOP Illegal encoder schedule num

Cause: An invalid sensor (encoder) schedule number was used within the specified tracking program instruction's SCH[] parameter.

Remedy: Check the \$LNSNRSCH[] system variable description for the range of valid sensor schedule numbers.

LNTK-011 STOP Illegal boundary set number

Cause: An illegal value was used within the specified tracking program instruction or within the program's associated schedule $\$LNSCH[i].\SEL_BOUND value (where 'i' is the FRAME number used within the DETAIL screen for the specified program).

Remedy: Check the value of the specified program instruction's BOUND[] parameter and the program's associated schedule $\$LNSCH[i].\SEL_BOUND value to make sure that they are one of the valid values listed under the description for this system variable.

LNTK-012 STOP Invalid input position

Cause: An invalid or uninitialized position was used within the specified tracking program instruction.

Remedy: Check the position (or position register) value for the specified tracking program instruction to make sure that it is a properly initialized, valid position.

LNTK-013 STOP Invalid trigger input value

Cause: An invalid or uninitialized value was used for the specified tracking program instruction's trigger value.

Remedy: Check the value of the program register used by the specified tracking program instruction.

LNTK-014 STOP Encoder/sensor not enabled

Cause: The tracking sensor (encoder) associated with the specified program (specified by $\$LNSCH[i].\TRK_ENC_NUM , where 'i' is the FRAME number used within the DETAIL screen for the specified program) must be enabled to perform this program instruction.

Remedy: Use the LINE enable instruction to enable the proper tracking sensor(encoder).

LNTK-015 STOP Invalid encoder trigger value

Cause: An invalid or uninitialized sensor (encoder) trigger value (specified by $\$LNSCH[i].\$TRIG_VALUE$, where 'i' is the FRAME number used within the DETAIL screen for the specified program) was found.

Remedy: Make sure that this value is properly set prior to either teaching path positions, or issuing programmed robot motion instructions.

LNTK-016 STOP Invalid input time

Cause: An invalid or uninitialized prediction time was used within the specified tracking program instruction.

Remedy: Check the prediction time being used for proper initialization.

LNTK-017 STOP Invalid input pointer

Cause: An invalid internal position input pointer was specified.

Remedy: Perform a COLD start of the system. (Cycle power) Notify FANUC Robotics if problem persists.

LNTK-018 STOP Invalid teach distance

Cause: An invalid or uninitialized teach distance value (specified by $\$LNSCH[i].\$TEACH_DIST$, where 'i' is the FRAME number used within the DETAIL screen for the specified program) was found.

Remedy: Make sure that this value is properly set prior to either teaching path positions, or issuing programmed robot motion instructions.

LNTK-019 STOP Invalid scale factor

Cause: An invalid or uninitialized scale factor value (specified by $\$LNSCH[i].\$SCALE$, where 'i' is the FRAME number used within the DETAIL screen for the specified program) was found.

Remedy: Make sure that this value is properly set prior to either teaching path positions, or issuing programmed robot motion instructions. NOTE: This value might NOT be equal to 0.0.

LNTK-020 STOP Invalid extreme position

Cause: An invalid or uninitialized extreme position value (specified by $\$LNSCH[i].\TCP_EXTRM , where 'i' is the FRAME number used within the DETAIL screen for the specified program) was found.

Remedy: Make sure that this value is properly set prior to either teaching path positions, or issuing programmed robot motion instructions. NOTE: A value of 1,000,000 (1.0e6) may be set to disable TCP extreme position checking.

LNTK-021 STOP Invalid track axis number

Cause: An invalid or uninitialized track axis number (specified by $\$LNSCH[i].\TRK_AXIS_NUM , where 'i' is the FRAME number used within the DETAIL screen for the specified program) was found.

Remedy: Make sure that this value is properly set to one of the valid values listed under the description for this system variable.

LNTK-022 STOP No tracking hardware

Cause: No tracking sensor hardware interface or improperly initialized system variables.

Remedy: Check tracking hardware setup and the values of $\$SCR.\ENC_TYPE and $\$SCR.\ENC_AXIS .

LNTK-023 STOP Bad tracking hardware

Cause: Bad tracking sensor hardware interface.
Remedy: Check all sensor hardware, cables, and connections.

LNTK-024 STOP Illegal encoder average

Cause: Illegal encoder average number.
Remedy: Use a valid encoder average number.

LNTK-025 STOP Illegal encoder multiplier

Cause: Illegal encoder multiplier number.
Remedy: Use a valid encoder multiplier number.

LNTK-026 STOP Encoder not enabled

Cause: Tracking encoder is not enabled.
Remedy: Enable the tracking encoder before reading its COUNT or RATE within the program.

LNTK-027 STOP Invalid data on LNTK stack

Cause: Invalid data was found on the tracking stack.
Remedy: Perform a COLD start of the system. (Cycle power.) Notify FANUC Robotics if problem persists.

LNTK-028 STOP LNTK stack underflow

Cause: The tracking stack attempted to read more data than was present.
Remedy: Perform a COLD start of the system. (Cycle power.) Notify FANUC Robotics if problem persists.

LNTK-029 STOP LNTK stack overflow

Cause: Too many tracking sub-processes are present. There is a limit to the number of tracking processes that can be called from other programs.
Remedy: Check to ensure sub-processes are not being called erroneously. Consider rewriting procedures so that fewer sub-processes are used.

LNTK-030 STOP Stack / header mismatch

Cause: The schedule number on the tracking stack did not match the schedule of the program it corresponds to.
Remedy: Perform a COLD start of the system. (Cycle power.) Notify FANUC Robotics if problem persists.

LNTK-031 STOP UFRAME must be zero

Cause: User frames cannot be used when tracking.
Remedy: Set \$MNUFRAMENUM[] to zero.

LNTK-032 STOP Conveyor resync failed

Cause: The conveyor was not resynchronized properly.
Remedy: Make sure the Tracking Schedule is properly initialized, the encoder is active, and all hardware is functioning properly.

MACR Error Codes (ID = 57)

MACR-001 WARN Can't assign to MACRO command

Cause: There are bad conditions to assign the macro.
Remedy: Is it Double definitions? Is the index over the range?

MACR-003 WARN Can't assign motn_prog to UK

Cause: It is impossible to assign the program with MOTION lock group to User Key (UK) button.
Remedy: Please remove the motion lock group from the program.

MACR-004 WARN Can't execute motn_prog by UK

Cause: It is impossible to execute the program with MOTION lock group by User Key(UK) button.
Remedy: Please remove the motion lock group from the program.

MACR-005 WARN Please enable teach pendant

Cause: It is possible to execute the program only when Teach pendant is enabled.
Remedy: Please change Teach pendant to be enabled.

MACR-006 WARN Please disable teach pendant

Cause: It is possible to execute the program only when Teach pendant is disabled.
Remedy: Please change Teach pendant to be disabled.

MACR-007 WARN The same macro type exists

Cause: The macro assign type has already exist.
Remedy: Please change the assign type to the other.

MACR-008 WARN Remote-cond isn't satisfied

Cause: This assign type is only enable at REMOTE condition.
Remedy: Please create REMOTE condition.

MACR-009 WARN The index is out of range

Cause: This assign index is out of range.
Remedy: Please change the assign index.

MACR-010 WARN This SOP button is disabled

Cause: This SOP button is disable for macro execution.
Remedy: Please change \$MACRSOPENBL.

MACR-011 WARN This UOP button is disabled

Cause: This UOP signal is disable for macro execution.
Remedy: Please change \$MACRUOPENBL.

MACR-012 WARN Number of DI+RI is over

Cause: The number of RI+DI is over the maximum number.
Remedy: Please release the other RI or DI setting.

MACR-013 WARN MACRO execution failed

Cause: Can't execute this MACRO.
Remedy: Please refer the HELP in the ALARM screen.

MACR-016 WARN The macro is not completed

Cause: The macro has aborted during the execution. The macro is executed from the first line at the next execution

MCTL Error Codes (ID = 6)**MCTL-001 NONE TP is enabled**

Cause: Teach pendant is enabled, and the motion control was not granted.

Remedy: Disable the teach pendant, and try the operation again.

MCTL-002 NONE TP is disabled

Cause: The teach pendant is disabled, and the motion control was not granted.

Remedy: Enable the teach pendant, and try the operation again.

MCTL-003 NONE system is in error status

Cause: Because the system is in error status, the motion control was not granted.

Remedy: Clear the error, and try the operation again.

MCTL-004 NONE motion is in progress

Cause: The motion is still in progress, and the motion control was not granted.

Remedy: Wait until the robot becomes the complete stop.

MCTL-005 NONE not in control of motion

Cause: The motion control was not granted because of some unknown reason.

Remedy: Clear the reason, and try the operation again.

MCTL-006 NONE TP has motion control

Cause: Because the teach pendant currently has the motion control, the motion control was not granted.

Remedy: Disable the teach pendant, and try the same operation again.

MCTL-007 NONE PROG has motion control

Cause: Because the program has the motion control, the motion control was not granted.

Remedy: Pause or abort the program, and try the same operation again.

MCTL-008 NONE Operator panel has motion control

Cause: Because the operator panel has the motion control, the motion control was not granted.

Remedy: Set the \$rmt_master system variable correctly, and try the operation again.

MCTL-009 NONE Other has motion control

Cause: Other device has the motion control, and the motion control was not granted.

Remedy: Set the \$rmt_master system variable correctly, and try the operation again.

MCTL-010 NONE Other than msrc is rel'ing

Cause: If you see this error, call our service representative.

MCTL-011 NONE Due to error processing

Cause: If you see this error, call our service representative.

MCTL-012 NONE subsystem code unknown

Cause: If you see this error, call our service representative.

MCTL-013 NONE ENBL input is off

Cause: ENBL input on the UOP is off.

Remedy: Set ENBL input ON.

MCTL-014 NONE Waiting for Servo ready

Cause: The motion control was not granted because servo was not up.

Remedy: Please wait for a few seconds until servo is up and ready.

MCTL-015 NONE Manual brake enabled

Cause: The motion control was not granted because manual brake control is enabled.

Remedy: Please disable the manual brake control.

MEMO Error Codes (ID = 7)

- MEMO-002 WARN Specified program is in use**
Cause: The specified program is editing or executing.
Remedy: Abort the specified program. Or select it once more after select another program.
- MEMO-003 WARN Specified program is in use**
Cause: The specified program is editing or executing.
Remedy: Abort the specified program. Or select it once more after select another program.
- MEMO-004 WARN Specified program is in use**
Cause: The specified program is editing or executing.
Remedy: Abort the specified program. Or select it once more after select another program.
- MEMO-006 WARN Protection error occurred**
Cause: The specified program is protected by user.
Remedy: Cancel the protection of the specified program.
- MEMO-007 WARN Invalid break number**
Cause: The specified break number does not exist.
Remedy: Specify the correct break number.
- MEMO-008 WARN Specified line no. not exist**
Cause: The specified line number does not exist in the specified or default program.
Remedy: Specify a correct line number.
- MEMO-010 WARN Program name error**
Cause: The specified program name is different from that of the P-code file.
Remedy: Specify the same program name.
- MEMO-013 WARN Program type is different**
Cause: The specified program type is different from that of the object being processed.
Remedy: Specify the same program type.
- MEMO-014 WARN Specified label already exists**
Cause: The specified label id already exists in the program.
Remedy: Specify another label number.
- MEMO-015 WARN Program already exists**
Cause: The specified program already exists in the system.
Remedy: Specify another program name. Or delete the registered program.
- MEMO-019 WARN Too many programs**
Cause: The number of the programs and routines exceeded the maximum possible number (3200).
Remedy: Delete unnecessary program or routine.
- MEMO-025 WARN Label does not exist**
Cause: Specified label does not exist.
Remedy: Set the index to an existing label.
- MEMO-026 WARN Line data is full**
Cause: The number of line data exceeded the maximum possible line number (65535).
Remedy: Delete unnecessary line data.
- MEMO-027 WARN Specified line does not exist**
Cause: The specified line data does not exist.
Remedy: Specify another line number.
- MEMO-029 WARN The line data can't be changed**
Cause: The specified line data can't be changed. The size of modified data is different from that of original data when replacing it.
Remedy: Specify another line number or the data of same size.
- MEMO-032 WARN Specified program is in use**
Cause: The specified program is editing or executing.
Remedy: Abort the specified program. Or select it once more after select another program.
- MEMO-034 WARN The item can't be changed**
Cause: The specified item is locked to change by system.
Remedy: Specify another item.

MEMO-038 WARN Too many programs

Cause: The number of the programs exceeded the maximum possible number.
Remedy: Delete unnecessary program.

MEMO-048 WARN Break point data doesn't exist

Cause: The specified break point data does not exist.
Remedy: Specify another break point.

MEMO-050 WARN Program does not exist

Cause: The specified program does not exist in the system.
Remedy: Specify another program or create the same program first.

MEMO-056 WARN Program does not exist

Cause: The specified program does not exist in the system.
Remedy: Specify another program or create the same program first.

MEMO-065 WARN Too many opened programs

Cause: Too many CALL instructions is used. The number of opened programs exceeded the maximum possible number(100).
Remedy: Abort the unnecessary programs. Or, remove unnecessary CALL instructions.

MEMO-071 WARN Position does not exist

Cause: The specified position data does not exist.
Remedy: Specify another position.

MEMO-072 WARN Position data already exists

Cause: Position data already exists in the specified position you want to move.
Remedy: Specify another position. Or, delete the data in the specified position.

MEMO-073 WARN Program does not exist

Cause: The specified program does not exist in the system.
Remedy: Specify another program or create the same program first.

MEMO-074 WARN Program type is not TPE

Cause: The operation can be apply only to TPE programs.
Remedy: Select the TPE program.

MEMO-075 WARN Program can't be used

Cause: The program is out of processing object program.
Remedy: Specify correct program.

MEMO-078 WARN Program can't be used

Cause: The program type that is out of processing object program is set.
Remedy: Specify the correct program type.

MEMO-080 WARN Protection error occurred

Cause: The specified program is protected by user.
Remedy: Cancel the protection of the specified program.

MEMO-081 WARN Specified program is in use

Cause: The specified program is editing or executing.
Remedy: Abort the specified program. Or select it once more after select another program.

MEMO-088 WARN Program does not exist

Cause: The specified position data does not exist.
Remedy: Specify another position.

MEMO-093 WARN Specified program is in use

Cause: The specified program is editing or executing.
Remedy: Abort the specified program. Or select it once more after select another program.

MEMO-098 WARN EOF occurs in file access

Cause: EOF occurs in file access. When P-code file was scanned, EOF occurs.
Remedy: The P-code data may be broken. Translate the specified KAREL program again. Then reload the P-code.

MEMO-099 WARN Program name is wrong

Cause: The program name length is different from that of the P-code data.
Remedy: Check the program name of the specified program.

MEMO-103 WARN Check sum error occurred

Cause: The specified data was broken. This is the internal error.
Remedy: Please call FANUC service center.

MEMO-104 WARN Program already exists

Cause: The specified program already exists in the system.
Remedy: Specify another program name. Or delete the registered program.

MEMO-112 WARN Break data already exists

Cause: The specified break point data already exists in the program.
Remedy: Specify another break point.

MEMO-113 WARN File access error

Cause: Not connect the port that have the program you want to load.
Remedy: Check the port setting and connected device.

MEMO-114 WARN Break point can't be removed

Cause: The break point data can not be overwritten. The program is protected by user or executing.
Remedy: Cancel the protection of the program. Or, abort the program.

MEMO-115 WARN Break point can't be removed

Cause: The break point data can not be removed. The program is protected by user or executing.
Remedy: Cancel the protection of the program. Or, abort the program.

MEMO-119 WARN Application data doesn't exist

Cause: The specified application data does not exist. The program does not correspond to the specified application.
Remedy: Specify another application data. Create the program in the current system again.

MEMO-120 WARN Application data doesn't exist

Cause: The specified application data does not exist. The program does not correspond to the specified application.
Remedy: Specify another application data. Create the program in the current system again.

MEMO-123 WARN Application data doesn't exist

Cause: The specified application data does not exist.
The program is not correspond to the specified application.
Remedy: Specify another application data. Create the program in the current system again.

MEMO-124 WARN Program version is too new

Cause: KAREL program version number is newer than that of the system.
Remedy: Translate the program with an older version of the Translator.

MEMO-125 WARN Program version is too old

Cause: KAREL program version number is older than that of the system.
Remedy: Translate the program with a newer version of the Translator.

MEMO-126 WARN No more available memory

Cause: Lack of the memory which can be used.
Remedy: Delete unnecessary programs.

MEMO-127 WARN Pos reference over 255 times

Cause: Reference of the same position exceeded the maximum count (256).
Remedy: Set new position ID for the referenced position.

MEMO-128 WARN %s parameters are different

Cause: A routine exists in memory with a different parameter definition than the routine in the PC file being loaded.
Remedy: Update the calling convention in the KAREL program being loaded or delete the obsolete routine from system memory.

MEMO-129 WARN System error

Cause: The data of the system been broken.
Remedy: Please power up again.

MEMO-130 SYSTEM Please power up again

Cause: The data of the system been broken.
Remedy: Please power up again.

MEMO-131 SYSTEM Please power up again

Cause: The data of the system been broken.
Remedy: Please power up again.

MEMO-132 WARN %s has been broken

Cause: The data of the program has been broken at the power fail recover.
Remedy: Delete the program and create it again. Please call FANUC service center.

MEMO-133 SYSTEM Please power up again

Cause: The data of the system been broken.
Remedy: Please power up again.

MEMO-134 WARN TPE program %s already exists

Cause: The TPE program which has the same name already exists.
Remedy: After delete the TPE program, load the specified KAREL program again.

MEMO-135 WARN Cannot create TPE program here

Cause: The TPE program cannot be created in this start mode.
Remedy: Select the function menu to change the start mode.

MEMO-136 WARN Cannot load P-code here

Cause: The KAREL program cannot be loaded in this start mode.
Remedy: Select the function menu to change the start mode.

MEMO-137 WARN Load at Control Start Only

Cause: Specified KAREL program cannot be loaded in this mode. Because the same name program has already been loaded at controlled start.
Remedy: Load the program at controlled start.

MEMO-138 WARN Delete at Control Start Only

Cause: Specified program has already been loaded at controlled start. Because of this, you can only delete the program at controlled start.
Remedy: Delete the program at controlled start.

MEMO-144 WARN Header size too big

Cause: The TPE header size specified is too big. Must be less than 256.
Remedy: Change size to range of 1-256. If necessary, use multiple header records.

MOTN Error Codes (ID = 15)

MOTN-000 WARN Unknown error (M000)

Cause: Internal system error.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-001 STOP Internal error in osmkpkt

Cause: Internal system error.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-002 STOP Internal error in ossndpkt

Cause: Internal system error.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-003 STOP Internal error in oswrtnbx

Cause: Internal system error.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-004 STOP Internal error in ossigflg

Cause: Internal system error.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-005 STOP Internal error in osclrflg

Cause: Internal system error.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-006 STOP Internal error in osrcvpkt**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-007 STOP Internal error in osredmbx**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-008 STOP Internal error in oswaiflg**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-009 STOP Internal error for single step**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-010 STOP Internal error in osathpkt**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-011 STOP Internal error in osdltpkt**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-012 STOP Invalid softpart MIR**Cause:** Invalid softpart MIR.**Remedy:** Make sure the correct basic motion softpart is installed**MOTN-013 STOP Invalid softpart SEG****Cause:** Invalid softpart SEG.**Remedy:** Make sure the correct basic motion softpart is installed.

MOTN-014 WARN unknown error (MO14)**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-015 WARN unknown error (MO15)**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-016 WARN unknown error (MO16)**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-017 STOP Limit error (G:%d^2, A:%x^3 Hex)**Cause:** Limit error.**Remedy:** Reteach the position out of limits.**MOTN-018 STOP Position not reachable****Cause:** Position not reachable Or near by singularity.**Remedy:** Reteach the position that is not reachable.**MOTN-019 WARN In singularity****Cause:** Position near by singularity.**Remedy:** Reteach the position that is near a singularity point.**MOTN-020 WARN Wristjoint warning****Cause:** Wrist joint warning.**Remedy:** Wrist joint warning.**MOTN-021 STOP No kinematics error****Cause:** No kinematics.**Remedy:** Use joint motion.**MOTN-022 STOP Invalid limit number****Cause:** Invalid limit number.**Remedy:** Set limit number correctly.**MOTN-023 STOP In singularity****Cause:** The position is near a singularity point.**Remedy:** Reteach the position that is near a singularity point.**MOTN-024 STOP Kinematics not defined****Cause:** Kinematics is not defined.**Remedy:** Define Kinematics.

MOTN-025 WARN unknown error (MO25)**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-026 WARN unknown error (MO26)**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-027 WARN unknown error (MO27)**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-028 WARN unknown error (MO28)**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-029 STOP unknown error (MO29)**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-030 STOP Internal error in MMGR:PEND**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-031 STOP Internal error in MMGR:ESEG**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-032 STOP Internal error in MMGR:PRSD**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-033 STOP Internal error in MMGR:GNL**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-034 STOP Internal error in MMGR_MMR**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-035 STOP Internal error in MMGR_MIR**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-036 STOP Internal error in MMGR:MSTR**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-037 STOP Internal error in MMGR:MDON**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-038 STOP Internal error in MMGR:CAN**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-039 STOP Internal error in MMGR:FCAN**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-040 STOP Internal error in MMGR:CAND**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-041 STOP Internal error in MMGR:PSTR**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-042 STOP Internal in MSSR**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-043 STOP Internal error in MMGR:EPKT**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-044 STOP Internal error in MMGR:ERR**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-045 STOP Internal error in pro. start**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-046 STOP Internal error in MMGR:LSTP**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-047 STOP Internal error in MMGR:PRST**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-048 STOP unknown error (MO48)**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-049 STOP Attempt to move w/o calibrated**Cause:** Robot not calibrated.**Remedy:** Calibrate the robot.**MOTN-050 STOP Invalid spdlim (G:%d^2 A:%x^3 H)****Cause:** Invalid joint speed limit.**Remedy:** Set \$SPEEDLIMJNT correctly.**MOTN-051 STOP Speed out of range (G:%d^2)****Cause:** Speed out of range.**Remedy:** Set speed correctly.**MOTN-052 STOP Jntvellim out of range (G:%d^2)****Cause:** Joint vel limit out of range.**Remedy:** Set \$JNTVELLIM correctly.**MOTN-053 STOP Internal planner error (G:%d^2)****Cause:** Internal Planner error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-054 STOP Uninitialized dest pos (G:%d^2)**Cause:** Uninitialized destination position.**Remedy:** Teach destination position.**MOTN-055 STOP Uninitialized via pos (G:%d^2)****Cause:** Uninitialized via position.**Remedy:** Teach via position.

MOTN-056 WARN Speed limits used (G:%d^2)

Cause: Speed limits used.

Remedy: This is just a notification that the command translational speed is not attained (before acceleration) due to rotational speed limits being applied. This message is also displayed if time-based motion is issued and the command time value cannot be attained due to rotational or translational speed limits being applied. If the slowdown is unacceptable, modify the program so that the orientation change is smaller (for non-time-based motion), or increase the segment time, or decrease the taught distance between points (for time-based motion).

MOTN-057 STOP Invalid mir (G:%d^2)

Cause: Invalid packet received by planner.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-058 STOP Invalid cancel request (G:%d^2)

Cause: Invalid cancel request received by planner.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-059 STOP Null segment received (G:%d^2)

Cause: Planner received null seg when not expecting one.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-060 STOP Uninitialized base vec (G:%d^2)

Cause: Uninitialized base vector in relative moves.

Remedy: Initialize base vector.

MOTN-061 STOP Uninitialized distance (G:%d^2)

Cause: Uninitialized distance in relative moves.

Remedy: Initialize distance.

MOTN-062 STOP Invalid position type (G:%d^2)

Cause: Invalid position type received by planner.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-063 STOP Position config change (G:%d^2)

Cause: Configuration mismatch.

Remedy: Reteach the destination position so that its configuration string matches the start position's configuration string.

MOTN-064 STOP Rs orientation error (G:%d^2)**Cause:** RS orientation planning error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-065 STOP AES orientation error (G:%d^2)**Cause:** AES orientation planning error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-066 STOP Degenerate circle (G:%d^2)**Cause:** Degenerate circle.**Remedy:** Reteach via and/or destination positions.**MOTN-067 STOP Ata2 error in circle (G:%d^2)****Cause:** Internal system error during circular planning.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-068 STOP Invalid Prgoverride (G:%d^2)**Cause:** Prgoverride is not within 0 to 100.**Remedy:** Set \$prgoverride within 0 to 100.**MOTN-069 STOP Error in mocmnd (G:%d^2)****Cause:** Internal error: planner received invalid mocmnd.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-070 STOP Error in motype (G:%d^2)**Cause:** Internal error: planner received invalid motype.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-071 STOP Error in termtpe (G:%d^2)**Cause:** Internal error: planner received invalid termtpe.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-072 STOP Error in segtermtype (G:%d^2)

Cause: Internal error: planner received invalid segtermtype.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-073 STOP Error in orientype (G:%d^2)

Cause: Internal error: planner received invalid orientype.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-074 STOP Error in speed (G:%d^2)

Cause: Speed is not within 0 to \$speedlim.

Remedy: Set speed within 0 to \$speedlim.

MOTN-075 STOP Error in rotspeed (G:%d^2)

Cause: Rotspeed is not within 0 to \$rotspeedlim.

Remedy: Set Rotspeed within 0 to \$rotspeedlim.

MOTN-076 STOP Error in contaxisvel (G:%d^2)

Cause: Contaxisvel is not within 0 to 100.

Remedy: Set contaxisvel to within 0 to 100.

MOTN-077 STOP Error in seg_time (G:%d^2)

Cause: Seg_time is negative.

Remedy: Set seg_time positive.

MOTN-078 STOP Error in accel_ovrd (G:%d^2)

Cause: Accel_ovrd greater than 500.

Remedy: Set accel_ovrd within 0 to 500.

MOTN-079 STOP Error in accu_num (G:%d^2)

Cause: Internal error: planner received invalid accu_num.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-080 STOP Via position required (G:%d^2)

Cause: Missing via position for circular motion.

Remedy: Teach via position.

MOTN-081 STOP Extended position error (G:%d^2)

Cause: Internal error: planner received invalid extended position representation.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-082 STOP Null mir pointer (G:%d^2)**Cause:** NULL MIR pointer.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-083 STOP Illegal SEG recvd (G:%d^2)**Cause:** Internal error: planner received segment belonging to another group.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-084 STOP Illegal CONSEG recvd (G:%d^2)**Cause:** Not used.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-085 STOP Error in gp_concurrent(G:%d^2)**Cause:** Internal error: planner received invalid mmr.gp_concurrent.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-086 STOP Not all CON_SEGs recvd(G:%d^2)**Cause:** Group motion: not all segments are received.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-087 STOP Utool change not allowed(G:%d^2)**Cause:** \$utool is changed before move.**Remedy:** Do not change \$utool for this move.**MOTN-088 STOP Not cartesian move (G:%d^2)****Cause:** Motype is not cartesian.**Remedy:** Must set motype to cartesian.**MOTN-089 STOP Segment not planned (G:%d^2)****Cause:** Internal plan error:seg in list not all planned.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-090 STOP MIR mismatch (G:%d^2)**Cause:** Internal plan error:mir mismatch.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-091 STOP Va orientation error (G:%d^2)**Cause:** Internal plan error:atan2 error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-092 STOP Extended not supported (G:%d^2)**Cause:** Extended axes not supported.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-093 STOP Internal PLAN blend err(G:%d^2)**Cause:** Internal plan error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-094 STOP Blend corner too big (G:%d^2)**Cause:** Not used.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-095 WARN Can't blend corner line:%d^5**Cause:** Warning, there is not enough distance to perform corner blending.**Remedy:** If corner blending is still required for the line shown reteach pos further apart**MOTN-096 STOP Cart rate not equal(G:%d^2)****Cause:** Intellitrak On: \$linear_rate and \$circ_rate must be equal.**Remedy:** Set \$linear_rate equal to \$circ_rate. cycle power**MOTN-097 WARN INTR overrun %d^3 (G:%d^2)****Cause:** Interpolator overrun.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-098 STOP Wrist singularity (G:%d^2)**Cause:** Not used.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-099 STOP INTR Fail to get MIRPKT (G:%d^2)**Cause:** Internal interpolator error:failed to receive mir when expecting one.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-100 STOP INTR Fail to get FDO (G:%d^2)**Cause:** Internal interpolator error:failed to receive fdo when expecting one.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-101 STOP MIR list is empty (G:%d^2)**Cause:** Internal interpolator error: mir list is empty when it shouldn't be.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-102 STOP SEG list is empty (G:%d^2)**Cause:** Internal interpolator error: seg list is empty when it shouldn't be.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-103 STOP Send ENB pkt fail (G:%d^2)**Cause:** Internal interpolator error: error in sending ENB packet.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-104 STOP Send DSB pkt fail (G:%d^2)**Cause:** Internal interpolator error: error in sending DSB packet.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-105 STOP Send TRG pkt fail (G:%d^2)

Cause: Internal interpolator error: error in sending TRG packet.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-106 STOP Process motion done (G:%d^2)

Cause: Internal interpolator error: process motion had completed without being restarted.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-107 STOP Bad filter type (G:%d^2)

Cause: Internal interpolator error: invalid filter type received.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-108 STOP INTR seglist error (G:%d^2)

Cause: Internal interpolator error: error in seg list management.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-109 STOP Internal INTR error (G:%d^2)

Cause: Internal interpolator error.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-110 STOP Use FINE in last L (G:%d^2)

Cause: Cannot replan joint motion in interpolator for this move.

Remedy: Use FINE in last L statement.

MOTN-111 WARN Can't switch filter(G:%d^2)

Cause: Warning message to indicate that switch filter cannot take place.

Remedy: This is just a notification. You do not have to do anything for this warning message.

MOTN-112 SABRT Increment move turn Mismatch

Cause: Incremental motion causes turn number mismatch.

Remedy: Change position to absolute position.

MOTN-113 WARN Robot not calibrated

Cause: Robot not calibrated.

Remedy: Calibrate the robot.

MOTN-114 WARN Servo is on (G:%d^2)

Cause: Servo is still on.

Remedy: Turn off servo.

MOTN-115 WARN Invalid brake mask (G:%d^2)

Cause: Invalid brake mask.
Remedy: Check brake mask.

MOTN-116 WARN Invalid solution (G:%d^2)

Cause: Invalid kinematics solution.
Remedy: Reteach position.

MOTN-117 WARN Robot not mastered (G:%d^2)

Cause: Robot not mastered.
Remedy: Master the robot. Refer to the *Setup and Operations Manual* specific to your application.

MOTN-118 WARN Robot in over travel (G:%d^2)

Cause: Robot in overtravel.
Remedy: Reset over travel jog the robot outside over travel position.

MOTN-119 WARN Servo is off (G:%d^2)

Cause: Robot servo is on.
Remedy: Turn off servo.

MOTN-120 WARN Invalid reference position (G:%d^2)

Cause: Invalid reference position.
Remedy: Check reference position.

MOTN-121 WARN Invalid config. string (G:%d^2)

Cause: Invalid config string.
Remedy: Reteach your config string.

MOTN-122 STOP Dfilter not empty (G:%d^2)

Cause: Internal system error.
Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-123 STOP Not enough node (G:%d^2)

Cause: Internal system error.
Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-124 STOP INTR:Bad Mirpkt req_code(G:%d^2)

Cause: Internal system error.
Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-125 STOP INTR got illegal pkt (G:%d^2)

Cause: Internal system error.
Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-126 STOP Can't init CH KPT (G:%d^2)**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-127 STOP Can't detach CH PKT (G:%d^2)**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-128 STOP Group mtn not supported(G:%d^2)**Cause:** Group motion not supported.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-129 STOP Local cond ptr conflict(G:%d^2)**Cause:** Conflict in local condition list pointers.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-130 STOP Non-empty local cond list(G:%d^2)**Cause:** Local condition list attached to SEG is not NULL.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-131 STOP In singularity**Cause:** Position near by singularity.**Remedy:** Reteach position that is near a singularity point.**MOTN-132 STOP Group circ not supported(G:%d^2)****Cause:** Group motion: circular motype for all groups not supported.**Remedy:** Reteach motype.**MOTN-133 WARN Time after limit used(G:%d^2)****Cause:** Local condition time after value is too big. System will use time after limit. This is just a warning.**Remedy:** No corrective action required.**MOTN-134 STOP Can not move path backward (G:%d^2)****Cause:** Backward path/subpath motion is not supported.**Remedy:** Remove backward command.**MOTN-135 STOP Last motype can't be circular (G:%d^2)****Cause:** Backward last node motype can not be circular.**Remedy:** Change last node motype.

-
- MOTN-136 STOP Illegal filter switch line:%d^5**
Cause: Cartesian filter to Joint filter transition supports only JOINT motype.
Remedy: Change motype to JOINT.
- MOTN-137 STOP No circular softpart (G:%d^2)**
Cause: The circular motion softpart is not loaded in the system.
Remedy: Load the circular softpart.
- MOTN-138 STOP No joint short motion SP (G:%d^2)**
Cause: Joint short motion softpart is not loaded in the system.
Remedy: Load joint short motion softpart.
- MOTN-139 STOP No cart short motion SP (G:%d^2)**
Cause: Cartesian short motion softpart is not loaded in the system.
Remedy: Load cartesian short motion softpart.
- MOTN-140 STOP No KAREL motion softpart (G:%d^2)**
Cause: The KAREL motion softpart is not loaded in the system.
Remedy: Load the KAREL motion softpart.
- MOTN-141 STOP No KAREL motion func. ptr (G:%d^2)**
Cause: The KAREL motion function pointer is not initialized or does not exist.
Remedy: Check that the KAREL Motion softpart has been loaded, and restart the controller.
- MOTN-142 STOP No Group Motion SP (G:%d^2)**
Cause: The Group Motion softpart is loaded and multi-group motion is specified.
Remedy: Check that the Group Motion softpart has been loaded, and restart the controller.
- MOTN-143 STOP No Motion Resume SP (G:%d^2)**
Cause: The Motion Resume softpart is loaded and path resume motion is specified.
Remedy: Check that the Motion Resume softpart has been loaded, and restart the controller.
- MOTN-144 STOP No joint Turbo Move SP (G:%d^2)**
Cause: Joint Turbo Move softpart is not loaded in the system.
Remedy: Load joint Turbo Move softpart.
- MOTN-145 STOP No cart Turbo Move SP (G:%d^2)**
Cause: Cartesian Turbo Move softpart is not loaded in the system.
Remedy: Load cartesian Turbo Move softpart.
- MOTN-146 STOP INTR can't replan major axis(G:%d^2)**
Cause: Mismatch in major axis turn number.
Remedy: Reteach position.
- MOTN-147 WARN L->J replan joint slowdown (G:%d^2)**
Cause: Linear motions ignore turn numbers. Therefore, when a joint motion follows several linear motions, the turn number might be mismatched, causing the robot to slow down.
Remedy: Change the current motion's motype to linear or change the previous motion's motype to joint. If the problem persists, re-teach the path.
- MOTN-148 WARN Can't move concurrently (G:%d^2)**
Cause: Two motion groups cannot synchronize with each other due to replanning of one group. This will cause slow down on both groups.
Remedy: If slow down is not acceptable, re-teach the path.
- MOTN-149 STOP CF:rotspeedlim exceeded line:%d^5**
Cause: CF:rotspeedlim exceeded.
Remedy: Set \$CF_PARAMGP[].[cf_framenum=1 or 2 and cycle power or reduce speed or use FINE in prev line.
- MOTN-300 STOP CD not support:Use CNT L:%d^5**
Cause: Term type CD is not supported.
Remedy: Change termtype FINE or CNT.
- MOTN-301 STOP Can't resume motion (G:%d^2)**
Cause: Can't resume motion.
Remedy: Abort and run program.

MOTN-302 WARN Corner speed slowdown L:%d^5

Cause: Corner speed slows down automatically because of robot constraint.

Remedy: If slow down is not acceptable, re-teach the path to provide a larger corner radius or increase the corner distance in the CD field.

MOTN-303 WARN Can't maintain CDist L:%d^5

Cause: Can't maintain corner distance because the node spacing is short or speed is high.

Remedy: Lengthen node spacing or reduce speed.

MOTN-304 WARN CS:Prog speed achieved L:%d^5

Cause: SPD value does not affect corner speed anymore.

Remedy: This is just a notification. You do not have to do anything for this warning message.

MOTN-305 WARN Can't maintain speed L:%d^5

Cause: Can't maintain program speed on the path because of robot constraint.

Remedy: This is just a notification. You do not have to do anything for this warning message.

MOTN-306 STOP Can't replan (G:%d^2, A:%x^3 Hex)

Cause: Resume motion cannot reach stop position Can't resume original path.

Remedy: Abort program and rerun.

MOTN-307 STOP Mismatch MMR (G:%d^2)

Cause: Internal system error. Can't resume original path.

Remedy: Abort program and rerun.

MOTN-308 WARN FINE termtyp used L:%d^5

Cause: Cannot generate a corner between two motions because of motion instruction. And CNT or CD is ignored.

Remedy: Use LOCK PREG instruction when PR[] is used for position or OFFSET instruction is used.

MOTN-309 WARN Circular speed reduced L:%d^5

Cause: Circular speed is reduced because of a robot constraint.

Remedy: Reduce the program speed not to display.

MOTN-310 STOP Pos. Cfg. change 2 (G:%d^2)

Cause: Configuration mismatch

Remedy: Reteach the destination position so that its configuration string matches the start position's configuration string.

MOTN-311 STOP Can't resume motion CJ (G:%d^2)

Cause: Can't resume motion on the original path.

Remedy: Abort and run program. Then, the resumed motion may not be on the original path.

MOTN-312 STOP Can't resume in single step CJ

Cause: Can't resume motion in single step mode.

Remedy: Abort program and rerun.

MOTN-313 STOP Can't resume motion CJ(2)

Cause: Can't resume motion on the original path.

Remedy: Abort and run program. Then, the resumed motion may not be on the original path.

MOTN-314 STOP unknown error (MO314)

Cause: Internal system error.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-315 STOP unknown error (MO315)

Cause: Internal system error.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-316 STOP unknown error (MO316)**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-317 STOP unknown error (MO317)**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-318 STOP unknown error (MO318)**Cause:** Internal system error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

MOTN-319 WARN CRC large orient change (G:%d^2)**Cause:** Small circle but large orientation change.**Remedy:** Reteach circular points.

PALT Error Codes (ID = 26)

- PALT-000 NONE UNUSED ERROR CODE(PLER00)**
- PALT-001 ABORT Inadequate register value**
Cause: Column/row/layer number is illegal.
Remedy: Please confirm palletizing register.
- PALT-004 ABORT Increment value is ill**
Cause: Increment value of palletizing config screen is illegal.
Remedy: Please correct increment value
- PALT-010 ABORT Route pattern unfound**
Cause: Route pattern unfound.
Remedy: Check route pattern in palletizing route pattern screen.
- PALT-024 ABORT Calculation error occurred**
Cause: Palletizing data is incomplete.
Remedy: Teach bottom point all? Teach rout point all?
- PALT-026 ABORT Cannot read/write to PL[]**
Cause: Cannot read/write to palletizing register.
Remedy: Check palletizing register index
- PALT-030 WARN Pallet number is over max**
Cause: Don't teach palletizing instruction over 16
Remedy: palletizing instruction don't teach more than 16 in this program, Please teach another program.
- PALT-031 WARN Can not be set FREE or INTER**
Cause: In FREE configuration, can not be set INTER over two directions
Remedy: In FREE configuration, can be set INTER only one direction (ROW, COLUMN or LAYER)

PRIO Error Codes (ID = 13)**PRIO-001 WARN Illegal iotype**

Cause: Port type specified is invalid
Remedy: Use one the port types defined in IOSETUP.KL

PRIO-002 WARN Illegal index

Cause: Port number is invalid or not presently assigned
Remedy: Correct the port number

PRIO-003 SYSTEM No memory available

Cause: Memory required for this operation is not available
Remedy: Delete Karel programs and/or variables to free memory

PRIO-004 WARN Too few ports on mod

Cause: There are not enough ports on the specified board or module to make the specified assignments
Remedy: Correct either the first port number or the number of ports

PRIO-005 WARN bad logical port no

Cause: The specified port number in an assignment is invalid; must be in the range 1..32767
Remedy: Correct the logical port number

PRIO-006 WARN bad log port number in asgt

Cause: The specified port number in an assignment is invalid; must be in the range 1..32767
Remedy: Correct the logical port number

PRIO-007 WARN no match in deassign call

Cause: Port being deassigned is not presently assigned
Remedy: Correct the port number

PRIO-008 WARN phys ports not found

Cause: Physical port being assigned to does not exist
Remedy: Correct the rack number, slot number, or port number

PRIO-009 WARN n_ports invalid

Cause: The number of ports in an assignment is invalid; must be in the range 1..128
Remedy: Correct the number of ports.

PRIO-010 WARN bad phys port number is asgt

Cause: Invalid physical port number in assignment request; must be greater than 1
Remedy: Correct the physical port number

PRIO-011 WARN asgt overlaps existing asgt

Cause: The logical port numbers being assigned overlap existing assignments
Remedy: Correct the first port number or number of ports

PRIO-012 WARN bad board num

Cause: The specified rack and/or slot number is invalid or refers to an unused rack/slot number.
Remedy: Correct the rack and/or slot number.

PRIO-013 WARN no aiseq for bd

Cause: An attempt was made to delete an analog input sequence which has not been defined.
Remedy: Check the rack and/or slot number.

PRIO-014 WARN ai seq too long

Cause: The specified analog input sequence is too long; sequence have from 1 to 15 ports numbers.
Remedy: Supply a sequence of an appropriate length.

PRIO-017 WARN I/O point not sim

Cause: Attempt to set input port that is not simulated
Remedy: Use the I/O menu to set the port simulated or avoid setting the port

PRIO-020 STOP SLC2-comm error %x, %x, %x, %x (hex)

Cause: Error in communications with process or modular I/O devices
Remedy: Check cabling between MAIN CPU PCB (JB-18 connector) and process I/O board and/or model A or B interface modules. Check power to remote Model A I/O racks and model B I/O interface modules. If corrections to these do not correct the problem, record the four numbers displayed in the error message and see the appropriate R-J2 Controller Maintenance manual for more detailed diagnostic information.

PRI0-023 WARN no ports of this type**Cause:** There are no ports of the specified type.**Remedy:** Change the port type, mount process I/O hardware with the required type of ports, or define ports (e.g., GIN or GOUT) ports of the specified type**PRI0-033 WARN PC interface init. fault****Cause:** PC interface board is bad or not installed.**Remedy:** Check for proper installation of PC interface. Check LED status on PC interface board. Refer to manual for possible cause.**PRI0-034 WARN PC interface genrl. fault %d****Cause:** PC interface board is faulted.**Remedy:** Check LED status on PC interface board. Refer to manual for possible cause.**PRI0-035 WARN PC interface serial fault %d****Cause:** PC interface serial link has failed.**Remedy:** Check LED status on PC interface board. Refer to manual for possible cause.**PRI0-063 WARN Bad IO asg: rack %d^1 slot %d^2****Cause:** One or more assignments to the process I/O board or module at specified rack and slot is invalid when the controller was turned on**Remedy:** Check the connections and power to the rack and that the module(s) are firmly installed. If the board or module has been permanently removed, use the CONFIG option in the DIO menu to delete the assignments.**PRI0-070 WARN PC interface option not loaded****Cause:** PC interface board is installed but the software option is not.**Remedy:** Install software option.**PRI0-072 WARN Pulse output is full****Cause:** Max of pulse output is 255 at the same time.**Remedy:** Check the count of pulse output.**PRI0-074 WARN Illegal pulse ID****Cause:** Specified pulse ID does not exist**Remedy:** Check the pulse ID.**PRI0-083 STOP I/O is not recovered****Cause:** I/O status in not recovered when hot start is enabled. I/O device or assignments are changed.**Remedy:** Initialize I/O by manual**PRI0-100 STOP Model B comm fault %srack:%d slot:%d****Cause:** Communication between Model B interface unit and DI/DO units is lost, or DI/DO unit power-off.**Remedy:** Check cable between Model B interface unit and DI/DO unit, or DI/DO unit power.

PROG Error Codes (ID = 3)

PROG-000 WARN Unknown error (PG0)

PROG-001 ABORT.L Invalid pointer is specified

Cause: This indicates an internal system error.

Remedy: Contact the FANUC Robotics Hot Line

PROG-002 ABORT.L Invalid task name is specified

Cause: The task name specified is invalid.

Remedy: Check the task name.

PROG-003 ABORT.L Invalid prog name is specified

Cause: The program name specified is invalid.

Remedy: Check the program name.

PROG-004 ABORT.L Invalid wait type is specified

PROG-005 WARN Program is not found

Cause: The specified program cannot be found.

Remedy: Check the program name.

PROG-006 WARN Line is not found

Cause: The specified line number cannot be found.

Remedy: Check the line number.

PROG-007 WARN Program is already running

Cause: The specified program is already being executed.

Remedy: Check the program name.

PROG-008 WARN In a rtn when creating a task

Cause: Execution cannot be started in sub-routine program.

Remedy: Check the line number.

PROG-009 WARN Line not same rtn as paused at

Cause: Resumption was attempted at a different line from the paused line.

Remedy: Check the line number.

PROG-010 WARN Not same prg as paused

Cause: Resumption was attempted in a different program from the paused one.

Remedy: Check the program name.

PROG-011 PAUSE.L Cannot get the motion control

Cause: Motion control cannot be obtained.

Remedy: Check the teach pendant enable switch and other running programs to determine who has motion control.

PROG-012 WARN All groups not on the top

Cause: There is paused motion later than motion that was attempted to resume.

Remedy: Resume the motion paused the last time.

PROG-013 WARN Motion is stopped by program

Cause: This motion was paused by the MOTION PAUSE instruction. Only the RESUME MOTION program instruction can resume the motion.

Remedy: Use RESUME MOTION instruction in the program.

PROG-014 WARN Max task number exceed

Cause: The number of programs you attempted to start exceeded the maximum number.

Remedy: Abort dispensable programs or perform a CTRL start and select PROGRAM INIT option to increase the number of tasks allowed.

PROG-015 WARN Cannot execute backwards

Cause: Backward execution cannot be used.

Remedy: Do not use backward execution at this point

PROG-016 WARN Task is not found

Cause: The specified task is not running or paused.

Remedy: Check the task name. The task name is always the name of the program that was run. The task name will not change even if the running program calls a routine from a different program.

PROG-017 WARN Task is not running

Cause: The specified task is not running.
Remedy: Check the task name.

PROG-018 ABORT.G Motion stack overflowed

Cause: Too many programs are paused.
Remedy: Resume or abort some programs.

PROG-019 WARN Ignore pause request

Cause: The request to pause the program was ignored.
Remedy: Change the NOPAUSE task attribute or use the KCL PAUSE command with the FORCE option.

PROG-020 WARN Task is already aborted

Cause: The specified program was already aborted.
Remedy: Check the program name.

PROG-021 WARN Ignore abort request

Cause: The request to abort the program was ignored.
Remedy: Change the NOABORT task attribute or use the KCL ABORT command with the FORCE option.

PROG-023 WARN Task is not paused

Cause: The specified program is not paused.
Remedy: Pause the program.

PROG-024 WARN Not have motion history

Cause: The motion path record is lost.
Remedy: Do not attempt backwards execution at this time.

PROG-025 WARN Cannot execute backwards

Cause: Backward execution cannot be used.
Remedy: Do not use backwards execution here.

PROG-026 WARN No more motion history

Cause: Backward execution cannot be used any more. The current line is on top of the memorized path.
Remedy: Do not use backwards execution here.

PROG-027 WARN Invalid task number

Cause: The task number specified is invalid.
Remedy: Check the task number.

PROG-029 WARN Buffer size is not enough

Cause: This indicates an internal system error.
Remedy: Please contact the FANUC Robotics hot line.

PROG-030 WARN Attribute is not found

Cause: The specified task attribute is not found.
Remedy: Check the attribute.

PROG-031 WARN Attribute is write protected

Cause: The specified task attribute is write protected.
Remedy: Do not try to change the attribute.

PROG-032 WARN Invalid value for attribute

Cause: The value for the specified attribute is invalid.
Remedy: Check the attribute value.

PROG-034 WARN Routine not found

Cause: The specified routine cannot be found.
Remedy: Check the routine name and verify it is loaded.

PROG-035 WARN Not locked the specified group

Cause: Motion control for the specified group cannot be locked.
Remedy: Check the teach pendant enable switch and other running programs to determine who has motion control.

PROG-036 WARN The length of trace array is 0

Cause: Not enough memory or the task attribute is not set correctly.
Remedy: Set the trace buffer length using the KCL SET TASK TRACELEN command.

PROG-037 WARN No data in the trace array

Cause: There is no execution record in memory.
Remedy: Turn on tracing using the KCL SET TRACE ON command.

PROG-039 WARN locked, but not get mctl

Cause: Motion control for the specified group was reserved, but it cannot be obtained.
Remedy: Check the teach pendant enable switch and other running programs to determine who has motion control.

PROG-040 PAUSE.L Already locked by other task

Cause: Motion control for the specified group was already reserved by another program.
Remedy: Check the other running programs to determine who has motion control.

PROG-041 WARN mctl denied because released

Cause: Motion control is released. The teach pendant currently has motion control. The robot cannot be started until motion control is obtained.
Remedy: Disable the teach pendant.

PROG-042 WARN Already released

Cause: Motion control was already released.
Remedy: If you had expected that the task may have already released the group, this may not be an error. Otherwise, check UNLOCK_GROUP usage.

PROG-043 WARN Already released by you

Cause: Motion control was already released by request of this program.
Remedy: If you had expected that the task may have already released the group, this may not be an error. Otherwise, check UNLOCK_GROUP usage.

PROG-044 WARN Arm has not been released yet

Cause: Motion control was not released yet.
Remedy: If you had expected that the task may have already locked the group, this may not be an error. Otherwise, check LOCK_GROUP usage.

PROG-045 WARN Other than requestor released

Cause: Motion control was already released by the request of another program.
Remedy: If you had expected that another task may have already released the group, this may not be an error. Otherwise, check UNLOCK_GROUP usage.

PROG-046 PAUSE.L TP is enabled while running (%s^7)

Cause: The teach pendant was enabled while the program is executing.
Remedy: Disable the teach pendant and resume the program.

PROG-047 PAUSE.L TP is disabled while running (%s^7)

Cause: The teach pendant was disabled while the program is executing.
Remedy: Enable the teach pendant and use shift-FWD to resume execution.

PROG-048 PAUSE.L Shift released while running (%s^7)

Cause: The shift key was released while the program is executing.
Remedy: Hold the shift key and press the FWD key to resume execution.

PROG-049 WARN Cannot release, robot moving

Cause: Motion control cannot be released because the robot is moving.
Remedy: Check the status of robot motion.

PROG-050 WARN Abort still in progress

Cause: The program is in the process of being aborted.
Remedy: Wait a second. If this error continues to occur, power down and power up the controller (COLD start).

PROG-051 WARN Cannot skip the return stmt

Cause: The specified lines to which a move was attempted exceed the number of lines in the program.
Remedy: Check the line number.

PROG-052 ABORT.L Process is aborted while executing

Cause: The user application task was forced to abort while the application is executing. The AMR may not have been completely processed.
Remedy: This requires no special action for the user.

PROG-053 ABORT.L User AX is not running

Cause: The user application task was not executed.
Remedy: Start the user application task before executing the application.

PWD Error Codes (ID = 31)**PWD-001 NONE Login (%s) Install**

Cause: A user with Install level access logged in.

Remedy: Status message only.

PWD-002 NONE Logout (%s) Install

Cause: A user with Install level access logged out.

Remedy: Status message only.

PWD-003 NONE Login (%s) Setup

Cause: A user with Setup level access logged in.

Remedy: Status message only.

PWD-004 NONE Logout (%s) Setup

Cause: A user with Setup level access logged out.

Remedy: Status message only.

PWD-005 NONE Login (%s) Program

Cause: A user with Program level access logged in.

Remedy: Status message only.

PWD-006 NONE Logout (%s) Program

Cause: A user with Program level access logged out.

Remedy: Status message only.

PWD-007 NONE Password Timeout (%s)

Cause: A user was logged out because of a password timeout.

Remedy: Log in, if required. Adjust the timeout value if it is too short.

PWD-008 NONE Create program %s.TP

Cause: A teach pendant program was created.

Remedy: Status message only.

PWD-009 NONE Delete program %s.TP

Cause: A teach pendant program was deleted.

Remedy: Status message only.

PWD-010 NONE Rename %s.TP as %s.TP

Cause: A teach pendant program was renamed.

Remedy: Status message only.

PWD-011 NONE Set %s.TP subtype from %s to %s

Cause: A teach pendant program subtype was changed. For example, a .TP program was changed to a Macro (.MR).

Remedy: Status message only.

PWD-012 NONE Set %s.TP comment

Cause: A teach pendant program comment was edited.

Remedy: Status message only.

PWD-013 NONE Set %s.TP group mask

Cause: The group mask of a teach pendant program was changed.

Remedy: Status message only.

PWD-014 NONE Set %s.TP write protect on

Cause: Write protection was enabled for the program. This helps prevent mistaken edits of the program.

Remedy: Status message only.

PWD-015 NONE Set %s.TP write protect off

Cause: Write protection was disabled for the program. The program can be edited.

Remedy: Status message only.

PWD-016 NONE Set %s.TP ignore pause on

Cause: The ignore pause feature was enabled for the program.

Remedy: Status message only.

PWD-017 NONE Set %s.TP ignore pause off

Cause: The ignore pause feature was disabled for the program.
Remedy: Status message only.

PWD-018 NONE Write line %d, %s.TP

Cause: A teach pendant program line was edited.
Remedy: Status message only.

PWD-019 NONE Delete line %d, %s.TP

Cause: A teach pendant program line was deleted.
Remedy: Status message only.

PWD-020 NONE Write pos %d, %s.TP

Cause: A teach pendant program position was recorded.
Remedy: Status message only.

PWD-021 NONE Delete pos %d, %s.TP

Cause: A teach pendant program position was deleted.
Remedy: Status message only.

PWD-022 NONE Renumber pos %d as %d, %s.TP

Cause: A teach pendant program position number was changed.
Remedy: Status message only.

PWD-023 NONE Set application data %s.TP

Cause: For some tool products, a teach pendant program may contain application related data. This message indicates the data has changed.
Remedy: Status message only.

PWD-024 NONE Delete application data %s.TP

Cause: For some tool products, a teach pendant program may contain application related data. This message indicates some data was deleted.
Remedy: Status message only.

PWD-025 NONE Load %s

Cause: The named file was loaded.
Remedy: Status message only.

PWD-026 NONE Load %s as Program %s

Cause: The named file was loaded. The program name may differ from the file name.
Remedy: Status message only.

PWD-027 NONE Edit %s Sch %d %s

Cause: A schedule was edited. Press HELP for more information.
Remedy: Status message only.

PWD-028 NONE Copy %s Sch %d to %d

Cause: The data in a schedule was copied to another schedule.
Remedy: Status message only.

PWD-029 NONE Clear %s Sch %d

Cause: The schedule was cleared, meaning the values were set to zero.
Remedy: Status message only.

PWD-030 NONE (%s to %s)%s

Cause: This message is used to provide detailed information for PWD-027. For example: PWD -027 Edit Weld Sch 1 Voltage PWD -030 (24.0 to 25.0) Volts
Remedy: Status message only.

PWD-031 WARN QUICK MENUS forced

Cause: The Operator password level does not have access to the FULL MENUS. Either a timeout occurred or a user logged out.
Remedy: Press the TP MENU hardkey and select SETUP PASSWORDS. Log in with either the Install, Setup, or Program password level. Press the TP FCTN hardkey and select QUICK/FULL MENUS to return to FULL MENU.

QMGR Error Codes (ID = 61)**QMGR-001 WARN Queue is full**

Cause: An attempt was made to add entry to a queue when the queue was full.

Remedy: Use GET_QUEUE to remove entries or use a larger value for queue size in the INIT_QUEUE call.

QMGR-002 WARN Queue is empty

Cause: An Attempt to use GET_QUEUE when there are no entries in the queue This is the normal result when no entries have been added or all have been removed by previous calls.

Remedy: No remedy is required.

QMGR-003 WARN Bad sequence no

Cause: A bad sequence_no value is used in an INSERT_QUEUE or DELETE_QUEUE call. The value may be less than 1 or greater than the sequence number of the last entry in the queue.

Remedy: Correct the value

QMGR-004 WARN Bad n_skip value

Cause: n_skip parameter in COPY_QUEUE call is less than zero

Remedy: Use zero or a positive value

ROUT Error Codes (ID = 17)**ROUT-022 PAUSE.G Bad index in ORD**

Cause: Incorrect number is specified for ORD builtin routine.

Remedy: Specify a number less than the string length

ROUT-023 PAUSE.G Bad index in SUBSTR

Cause: Incorrect number is specified for SUBSTR builtin routine.

Remedy: Specify a number less than the string length.

ROUT-024 PAUSE.G SUBSTR length less than 0

Cause: Negative number is specified for length argument for SUBSTR builtin routine.

Remedy: Specify a positive number.

ROUT-025 ABORT.G Illegal semaphore number

Cause: Incorrect number is specified for semaphore id.

Remedy: Specify a number between 1 and 32.

ROUT-026 WARN Illegal group number

Cause: Invalid group number is specified.

Remedy: Specify existing group number.

ROUT-027 WARN String size not big enough

Cause: Specified string variable does not have enough room to hold the return data.

Remedy: specify larger size string variable.

ROUT-028 ABORT.G Illegal file attribute number

Cause: Incorrect file attribute id was specified.

Remedy: Specify correct file attribute id.

ROUT-029 ABORT.G Illegal file attribute value

Cause: Incorrect file attribute value was specified.

Remedy: Specify correct attribute value.

ROUT-030 WARN Non existent register number

Cause: A non-existent register number is specified.

Remedy: Specify a correct register number

ROUT-031 WARN Illegal register type

Cause: Incorrect register type is specified.

Remedy: Specify the correct register type for the attempted operation.

ROUT-032 ABORT.G Position type mismatch

Cause: Position type is not correct for the operation.

Remedy: Specify correct position type.

ROUT-033 ABORT.G Illegal attribute type

Cause: Illegal attribute id was specified.

Remedy: Specify correct attribute id

ROUT-034 WARN Not a TPE program

Cause: A non-tpe is specified.

Remedy: Specify a program name other than a Karel program.

ROUT-035 WARN Value is out of range

Cause: The specified value is out of range.

Remedy: Specify a value within the range.

ROUT-036 ABORT.G Illegal port id value

Cause: Incorrect port id was used

Remedy: Specify correct port id.

ROUT-037 ABORT.G Bad TPE header size

Cause: Value used in SET_HEAD_TPE for bfr_size size is invalid.

Remedy: Use buffer size in the range 1-255

ROUT-038 PAUSE.G Uninitialized TPE position

SCIO Error Codes (ID = 25)

SCIO-016 WARN This option does not exist

SCIO-020 WARN LBL[%d] exists in line %d:

Cause: This label number exists in another line.

Remedy: Please select another label number.

SENS Error Codes (ID = 58)

- SENS-000 SYSTEM Unknown error**
Cause: System internal error
Remedy: Notify FANUC Robotics.
- SENS-001 STOP.G Hardware error occurred**
Cause: During data reception, parity, overrun and framing errors occurred.
Remedy: Check that the communication setting between the robot and sensor sides is not wrong.
- SENS-002 STOP.G DSR off when transmission**
Cause: An attempt was made for data transmission, but DSR signal at the sensor side is in OFF-state.
Remedy: Check the specification and connection of cable connecting the robot and sensor.
- SENS-003 STOP.G Undefined TCC received**
Cause: The undefined TCC was received from the sensor.
Remedy: Check the data sent from the sensor.
- SENS-004 STOP.G Invalid software parity**
Cause: BCC of the received data from the sensor is wrong.
Remedy: Check the data sent from the sensor.
- SENS-005 STOP.G Invalid data format**
Cause: The format of the received data is wrong.
Remedy: Check the data sent from the sensor.
- SENS-006 STOP.G Response time over**
Cause: The answer from the sensor is not received within the allowable time.
Remedy: Check that the sensor side does not stop due to an error, for example.
- SENS-007 STOP.G Interval time over**
Cause: The interval of characters sent from the sensor exceeded the allowable time.
Remedy: Check that the sensor side does not stop due to an error, for example.
- SENS-008 STOP.G Calculate matrix error**
Cause: Calculating transform matrix is impossible.
Remedy: Check the compensation data sent from the sensor.

SRVO Error Codes (ID = 11)**SRVO-001 SERVO Operator panel E-stop**

Cause: The operator panel emergency stop push button is pressed.

Remedy: Twist the operator panel emergency stop push button clockwise to release. Press reset.

SRVO-002 SERVO Teach pendant E-stop

Cause: The teach pendant emergency stop push button is pressed.

Remedy: Twist the teach pendant emergency stop push button clockwise to release. Press reset.

SRVO-003 SERVO Deadman switch released

Cause: The teach pendant deadman switch is released while the teach pendant is enabled.

Remedy: Press teach pendant deadman switch. Press reset.

SRVO-004 SERVO Fence open

Cause: FENCE1 and FENCE2 circuit open on EMG Control PCB.

Remedy: Determine the cause of FENCE1 and FENCE2 open circuit and correct. Press reset.

SRVO-005 SERVO Robot overtravel

Cause: A Robot overtravel limit switch, is pressed.

Remedy: To determine which axis is overtraveled:

1. Press MENUS.
2. Select MANUAL FCTNS.
3. Press F1, [TYPE].
4. Select OT_RELEASE Menu. The axis that is overtraveled will display TRUE in either OT_MINUS or OT_PLUS.
5. Move the cursor to the OT PLUS or OT MINUS value of the axis in overtravel.
6. Press F2, RELEASE. The value of the overtraveled axis should change back to FALSE.
7. Press and hold down the SHIFT key until you have completed steps a through d.
 - a. Press RESET and wait for servo power.
 - b. Press COORD until you select the JOINT coordinate system.
 - c. Continuously press and hold the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.
 - d. Jog the overtraveled axis off the overtravel switch. When you have finished jogging, you can release the SHIFT key.
8. Turn the teach pendant ON/OFF switch to OFF and release the DEADMAN switch.
9. Check CRM1 connection on axis control PCB if the robot is not in an actual overtravel condition. **NOTE:** If you accidentally release the SHIFT key during this procedure, you will need to repeat Step 7.

SRVO-006 SERVO Hand broken

Cause: The hand broken (*HBK) robot input is asserted.

Remedy: If using *HBK input, determine the cause of the error and correct. If not, check the position of the *HBK jumper on the axis control PCB; if on side A, *HBK is checked, if on side B, *HBK is not checked. *HBK originates on the Axis Control PCB.

SRVO-007 SERVO External emergency stops

Cause: The external emergency stop push button is pressed.

Remedy: If using external emergency stop, clear source of fault, and press reset. If not, check wiring at EMGIN1, EMGIN2, and EMGINC on EMG Control PCB. Check for 100 VAC input to the EMG Control PCB.

SRVO-008 SERVO Brake fuse blown

Cause: The brake fuse is blown on the EMG Control PCB. The FALM light on the EMG Control PCB should also be lit.

Remedy: Replace fuse on EMG Control PCB. Also see SRVO-018 Brake abnormal.

SRVO-009 SERVO Pneumatic pressure alarm

Cause: The pneumatic pressure (PPABN) robot input is asserted.

Remedy: If using pneumatic pressure input clear source of fault, press reset. If pneumatic pressure is not used set \$PARAM_GROUP[x].\$PPABN_ENBL system variable to FALSE. PPABN originates on the Axis Control PCB.

SRVO-010 SERVO Belt broken

Cause: The belt broken robot digital input (RDI7) is asserted.

Remedy: If using belt broken detection, clear source of fault, press reset. Robot inputs/outputs originate on the Axis Control PCB. Check system variable \$PARAM_GROUP.\$BELT_ENABLE.

SRVO-011 SERVO TP released while enabled

Cause: Teach pendant was disconnected while it was enabled.

Remedy: Re-connect the teach pendant, disable the teach pendant, and then disconnect the teach pendant. Note that if the teach pendant emergency stop is pressed when disconnecting the teach pendant, it will be necessary to re-connect to clear the SRVO-002 alarm.

SRVO-012 SERVO Power failure recovery

Cause: Normal power on (hot start).

Remedy: This is just a notification. You do not have to do anything for this warning message.

SRVO-013 SYSTEM Srvo module config changed

Cause: The configuration of digital servo modules has changed.

Remedy: Re-configure system with new digital servo module changes. Cycle power.

SRVO-014 WARN Fan motor abnormal

Cause: Card rack fan motor overheat.

Remedy: Check and/or replace defective fan.

SRVO-015 SERVO System over heat

Cause: Overheat sensor on backplane closed.

Remedy: The cabinet overheat sensor is located on the backplane. If the internal cabinet temperature is greater than 65 degrees Centigrade, check the cabinet fans for proper operation. Replace the backplane if cabinet temperature is within specification.

SRVO-016 SERVO Cooling water volume drop

Cause: Cooling water volume dropped (L1000 only).

Remedy: Determine the cause of the problem and repair.

SRVO-017 SERVO No robot internal mirror

Cause: No robot internal mirror (L1000 only).

Remedy: Determine the cause of the problem and repair.

SRVO-018 SERVO Brake abnormal

Cause: The FET current for brake exceeded the specification.

Remedy: Check brake for zero or abnormally low impedance. Then check the brake cable. Then check 200VAC. Then check servo amplifier or emergency stop control PCB if brake ports are used.

SRVO-019 SERVO SVON input

Cause: SVON (Servo ON/OFF switch) input asserted.

Remedy: Determine the cause to input SVON and repair.

SRVO-020 SERVO SRDY off (TP)

Cause: The teach pendant cable is disconnected or a momentary break occurred in any one of the TP emergency stop circuits; TP emergency stop, deadman, or fence.

Remedy: Check the teach pendant cable and connections.

SRVO-021 SERVO SRDY off/Door open (G:%d A:%d)

Cause1: The axis control asserts *MCON signal to servo amplifier, the servo amplifier asserts *DRDY. If *DRDY can not be asserted and the servo amplifier can not determine the problem, this alarm occurs.

Remedy1: Check the voltage at 100A and 100B, if this voltage is below 85V, determine the cause and repair. Check the cables and connections between servo amplifier and axis control PCB. Replace Servo Interface (SIF) module on axis control PCB. Replace the servo amplifier.

Cause2: The controller door is open.

Remedy2: Close the controller door.

SRVO-022 SERVO SRDY on (Group:%d Axis:%d)

Cause: The axis control asserts *MCON signal to servo amplifier, the servo amplifier asserts *DRDY. If *DRDY is already asserted, this alarm occurs.

Remedy: Check the cables and connections between servo amplifier and axis control PCB. Replace Servo Interface (SIF) module on axis control PCB. Replace the servo amplifier.

SRVO-023 SERVO Stop error excess(G:%d A:%d)

- Cause:** When the robot is at rest servo error is too big, greater than acceptable stop error tolerance.
- Remedy:** If the robot is loaded beyond specification, the torque necessary to decelerate a overloaded motor may cause this alarm to occur. Check the three phase input to the servo amplifier for voltage within specification; 170 – 253 VAC. Also, check for balanced voltage between all three phases. Check the cables and connections between servo amplifier and axis control PCB. Replace the Servo Interface (SIF) module on axis control PCB. Replace the servo amplifier.

SRVO-024 SERVO Move error excess(G:%d A:%d)

- Cause:** The servo error is too big when the the robot is moving, or if the robot moves when it is supposed to be stopped. The servo error in this case is greater than acceptable move error tolerance
- Remedy:** Same as SRVO-023 Stop error excess.

SRVO-025 SERVO Motn dt overflow (G:%d A:%d)

- Cause:** The motion command exceeded specification. Internal motion error.
- Remedy:** Perform a cold start:
1. Turn off the robot.
 2. On the teach pendant, press and hold the SHIFT and RESET keys.
 3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

SRVO-026 WARN Motor speed limit(G:%d A:%d)

- Cause:** Motor can not rotate as fast as the calculated speed required for the current motion.
- Remedy:** This is just a notification. However, you should attempt to eliminate this error and not repeat the circumstances that led up to it.

SRVO-027 WARN Robot not mastered(Group:%d)

- Cause:** System variable \$master_done set FALSE.
- Remedy:** Master robot.

SRVO-028 STOP Servo reset (Group:%d)

- Cause:** No longer occurs.
- Remedy:** N/A

SRVO-029 STOP Robot calibrated (Group:%d)

- Cause:** No longer occurs.
- Remedy:** N/A

SRVO-030 SERVO Brake on hold (Group:%d)

- Cause:** This alarm occurs when HOLD is pressed with brake on hold option turned on.
- Remedy:** This is just a notification. You do not have to do anything for this warning message.

SRVO-031 SERVO User servo alarm (Group:%d)

- Cause:** User servo alarm posted by the user.
- Remedy:** This is just a notification. You do not have to do anything for this warning message.

SRVO-032 STOP Force follow-up end (Grp:%d)

- Cause:** No longer occurs.
- Remedy:** N/A

SRVO-033 WARN Robot not calibrated(Grp:%d)

- Cause:** System variable \$calibrate set FALSE.
- Remedy:** Calibrate the robot.

SRVO-034 WARN Ref pos not set (Group:%d)

- Cause:** Reference position has not been set when quick mastering.
- Remedy:** Quick mastering may not be possible. Fixture or zero master.

SRVO-035 WARN Joint speed limit(G:%d A:%d)

- Cause:** Joint can not rotate as fast as the calculated speed required for the current motion.
- Remedy:** This is just a notification. However, every attempt should be made to eliminate this error.

SRVO-036 SERVO Inpos time over (G:%d A:%d)

Cause: Robot is not in position for the specified period of time.

Remedy: Check if the robot is loaded beyond specification. The torque necessary to decelerate a overloaded motor may cause this alarm to occur. Check the three phase input to the servo amplifier for voltage within specification; 170 – 253 VAC. Also, check for balanced voltage between all three phases. Check the cables and connections between servo amplifier and axis control PCB. Replace the Servo Interface (SIF) module on axis control PCB. Replace the servo amplifier.

SRVO-037 SERVO IMSTP input (Group:%d)

Cause: IMSTP (immediate stop) UOP input asserted.

Remedy: If using UOP, determine the cause and repair. If not using UOP, select the I/O menus and zero UOP mapping.

SRVO-038 SERVO Pulse mismatch (G:%d A:%d)

Cause: Pulse counts at power down and at power up are mismatch

Remedy: This feature is only available after core software version V3.06P. If your software version is V3.06P or V3.06PA set \$MCR.\$SPC_RESET true from the teach pendant and remaster the robot. If your software version is V3.06PB or greater, press RES_PCA (F3) softkey in the SYSTEM Master/Cal window, and remaster the robot. If this problem occurs repeatedly, replace the pulse coder.

SRVO-039 SERVO Motor speed excess(G:%d A:%d)

Cause: CMC cannot work because the calculated motor speed exceeded specification

Remedy: Reduce the motion speed or disable CMC.

SRVO-040 WARN Mastered at mark pos(G:%d)

Cause: Zero position master is done with mark position (not with zero position).

Remedy: This message is only for S-420iR. S-420iR has the mark at non-zero position for J2 and J3. Zero position master is not done with zero pos for S-420iS. Confirm the position of each axis to be at mark position. If the robot is not S-420iR, \$SCR_GRP.\$robot_model may be wrong. Set correct \$SCR_GRP.\$robot_model.

SRVO-041 SERVO MOFAL alarm (Grp:%d Ax:%d)

Cause: The motion command after the ramping algorithm in servo software exceeded one word.

Remedy: Internal motion error. Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

SRVO-042 SERVO MCAL alarm(Group:%d Axis:%d)

Cause: The servo amplifier magnetic contactor (MCC) is welded closed.

Remedy: If this alarm occurs with a SRVO-049 OHAL1, turn off the controller power for fifteen seconds and turn on again. Check the cable between the servo amplifier and axis control PCB. Replace the servo amplifier.

SRVO-043 SERVO DCAL alarm(Group:%d Axis:%d)

Cause: The regenerative energy produced by the motor, exceeded specification.

Remedy: If a 4 is indicated on the servo amplifier LED, a DCSW condition exists. DCSW alarm occurs when the regenerative transistor is on for one second or longer. To repair a DCSW, replace the servo amp. If problem persists, the load of the robot may exceed the specification. If a 5 is indicated on the servo amplifier LED, a DCOH condition exists. DCOH alarm occurs when the regenerative resistor overheats and is sensed by the thermostat. The average regenerative energy is excessive, relax the operating conditions. If using a separate regenerative discharge unit, check the wiring or replace.

SRVO-044 SERVO HVAL alarm(Group:%d Axis:%d)

Cause: The DC voltage on the main power circuit of the servo amplifier exceeded specification.

Remedy: Check the three phase voltage to the servo amplifier input. It should not exceed 253 VAC phase-to-phase. If the load on the robot exceeds the specification, this alarm could occur. If using a separate regenerative discharge unit, check the wiring or replace. Replace the servo amplifier. For auxiliary axes, the operating condition (duty cycle) may not be appropriate for the specification of the motor or amplifier. If the duty cycle can not be reduced, select a larger motor and amplifier.

SRVO-045 SERVO HCAL alarm(Group:%d Axis:%d)

Cause: The current in the main power circuit of the servo amplifier exceeded specification.

Remedy: Disconnect the motor power wires from the servo amplifier and turn on power. If an HCAL occurs, replace the transistor module or servo amplifier. Measure the resistance between GND and U, V, W individually on the cable terminals. If shorted, determine if the cable or motor is bad. Check the resistance between U-V, V-W, and W-U using a measuring instrument sensitive enough to detect small resistances at the cable terminations. If the resistances are the same replace the servo amplifier. If the resistances are different, determine if the cable or motor is bad. If the problem persists, replace the SIF module on the axis control for the defective axis.

SRVO-046 SERVO OVC alarm (Group:%d Axis:%d)

Cause: The average current calculated by the servo software exceeded specification.

Remedy: Make sure the robot is not loaded beyond specification. Check input power to the servo amplifier. It should be greater than 170 VAC phase-to-phase. Replace SIF module on the axis control PCB.

SRVO-047 SERVO LVAL alarm(Group:%d Axis:%d)

Cause: The DC voltage on the main power circuit of the servo amplifier is lower than the specification even though MCC is on.

Remedy: If a 2 is indicated on the servo amplifier LED, the 5 VDC is 4.6 volts. volts or less. Check input power to the servo amplifier. It should be greater than 170 VAC phase-to-phase. Replace the servo amplifier if the input power is correct. If a 3 is indicated on the servo amplifier LED, the main power is too low. Check input power to the servo amplifier. It should be greater than 170 VAC phase-to-phase. Replace the servo amplifier if the input power is correct. Check to make the circuit breaker is not off. If a 7 is indicated on the servo amplifier LED, MCC is welded closed.

SRVO-048 SERVO MOH alarm (Group:%d Axis:%d)

Cause: Never occurs on R-J2 Robot.

Remedy: None applicable.

SRVO-049 SERVO OHAL1 alarm (Grp:%d Ax:%d)

Cause: The servo amplifier overheated.

Remedy: If the robot is overloaded or the duty cycle exceeds specification, this alarm occurs. Check regenerative discharge transistor Q1. Check the thermostat on the servo amplifier after the servo amplifier has cooled. It should not be open. If the problem persists, replace the servo amplifier. Check controller cabinet fans for blocked filters, clean if necessary.

SRVO-050 SERVO Collision Detect alarm (G:%d A:%d)

Cause: The servo software detected a disturbance torque that was too high, and tripped a collision detection alarm.

Remedy: Reset the robot by using the teach pendant reset and JOG the robot away from any obstruction. If no collision, the load on the robot may exceed the specification. Check input power to the servo amplifier. It should be greater than 170 VAC phase-to-phase. Also check the voltage between U-V, V-W, and U-W. Each should measure the same (~210VAC).

SRVO-051 SERVO CUER alarm(Group:%d Axis:%d)

Cause: The feedback current is abnormal.

Remedy: Replace the SIF module on the axis control PCB. Replace the servo amplifier.

SRVO-052 WARN Discharge excess (Amp:%d)

Cause: NOT used

Remedy:

SRVO-053 WARN Disturbance excess(G:%d A:%d)

Cause: Disturbance estimated in the software exceed the threshold value. There is the possibility that the load held in the wrist exceed the robot specification.

Remedy: Reduce the load into the robot spec.

SRVO-054 SERVO DSM memory error (DS:%d)

Cause: The DSP module program memory is defective.

Remedy: Replace the DSP module.

SRVO-061 SERVO CKAL alarm(Group:%d Axis:%d)

Cause: The clock for the rotation counter in the pulse coder is abnormal.

Remedy: If this alarm occurs along with a SRVO-068 DTERR, SRVO-069 CRCERR, or SRVO-070 STBERR, disregard this alarm and refer to the other three alarm remedies. Replace the pulse coder or motor and master the robot.

SRVO-062 SERVO BZAL alarm(Group:%d Axis:%d)

Cause: The battery voltage for the pulse coder is zero volts.

Remedy: If this alarm occurs along with a SRVO-068 DTERR, SRVO-069 CRCERR, or SRVO-070 STBERR, disregard this alarm and refer to the other three alarm remedies. If SRVO-047 LVAL occurs before this alarm, batteries are drained to zero. Replace the pulse coder batteries and master the robot. If no SRVO-047 LVAL occurs before, check the battery cables to the motors. You may have to reset the pulse coder to clear this alarm. Refer to the SPC_RESET procedure in SRVO-038 and cycle the controller power. The controller may come back up with a SRVO-038 and require a second SPC_RESET.

SRVO-063 SERVO RCAL alarm(Group:%d Axis:%d)

Cause: The built-in rotation counter on the pulse coder is abnormal.

Remedy: If this alarm occurs along with a SRVO-068 DTERR, SRVO-069 CRCERR, or SRVO-070 STBERR, disregard this alarm and refer to the other three alarm remedies. Replace the pulse coder or motor and master the robot.

SRVO-064 SERVO PHAL alarm(Group:%d Axis:%d)

Cause: The relationship between the analog signals on the pulse coder are abnormal.

Remedy: If this alarm occurs along with a SRVO-068 DTERR, SRVO-069 CRCERR, or SRVO-070 STBERR, disregard this alarm and refer to the other three alarm remedies. Replace the pulse coder or motor and master the robot.

SRVO-065 WARN BLAL alarm(Group:%d Axis:%d)

Cause: The pulse coder batteries are low.

Remedy: If this alarm occurs along with a SRVO-068 DTERR, SRVO-069 CRCERR, or SRVO-070 STBERR, disregard this alarm and refer to the other three alarm remedies. Replace the pulse coder batteries while controller power is turned on.

SRVO-066 SERVO CSAL alarm(Group:%d Axis:%d)

Cause: The pulse coder ROM checksum data are abnormal.

Remedy: If this alarm occurs along with a SRVO-068 DTERR, SRVO-069 CRCERR, or SRVO-070 STBERR, disregard this alarm and refer to the other three alarm remedies. Replace the pulse coder or motor and master the robot.

SRVO-067 SERVO OHAL2 alarm (Grp:%d Ax:%d)

Cause: The pulse coder or motor overheated.

Remedy: If this alarm occurs along with a SRVO-068 DTERR, SRVO-069 CRCERR, or SRVO-070 STBERR, disregard this alarm and refer to the other three alarm remedies. If the load on the robot or duty cycle exceeds the specification, this alarm will occur. Allow the motor to cool. If the alarm stills occurs, replace the pulse coder or motor.

SRVO-068 SERVO DTERR alarm (Grp:%d Ax:%d)

Cause1: The axis control PCB sent the request signal, but did not receive serial data from the pulse coder.

Remedy1: Check pulse coder cables. Replace the SIF module on the axis control PCB. Replace the DSM module on the axis control PCB. Replace the pulse coder. If a serial pulse coder is plugged into a line tracking port, this alarm will occur. Check axis control PCB hardware configuration.

Cause2: The memory card interface (with an installed memory card) is plugged into the ER-2 printed circuit board while the controller is running and the ER-2 board is connected to the PLC.

Remedy2: Plug the memory card interface into a different slot. Or, connect the memory card interface directly to the backplane without using a printed circuit board.

SRVO-069 SERVO CRCERR alarm (Grp:%d Ax:%d)

Cause: The serial data from the pulse coder changed during communication to the axis control PCB.

Remedy: Check pulse coder cables. Make sure the cable shields are grounded. Replace the SIF module on the axis control PCB. Replace the DSM module on the axis control PCB. Replace the axis control PCB. Replace the pulse coder.

SRVO-070 SERVO STBERR alarm (Grp:%d Ax:%d)

Cause: The communication stop and start bits are abnormal.

Remedy: Check pulse coder cables. Replace the SIF module on the axis control PCB. Replace the DSM module on the axis control PCB. Replace the pulse coder.

SRVO-071 SERVO SPHAL alarm (Grp:%d Ax:%d)

Cause: The feedback velocity exceeds specification.

Remedy: If this alarm occurs with another pulse coder alarm, refer to the remedy of the other alarm first. If no other alarms, the robot load may exceed the specification. If the load is within specification, replace the serial pulse coder or motor.

SRVO-072 SERVO PMAL alarm(Group:%d Axis:%d)

Cause: The interpolation circuits of the pulse coder are abnormal.

Remedy: If this alarm occurs along with a SRVO-068 DTERR, SRVO-069 CRCERR, or SRVO-070 STBERR, disregard this alarm and refer to the other three alarm remedies. Replace the pulse coder and master the robot.

SRVO-073 SERVO CMAL alarm(Group:%d Axis:%d)

Cause: Incorrect position data detected in the pulse coder, or abnormal pulse coder data caused by noise.

Remedy: If this alarm occurs along with a SRVO-068 DTERR, SRVO-069 CRCERR, or SRVO-070 STBERR, disregard this alarm and refer to the other three alarm remedies. Master the robot. Check and strengthen the shield of the pulse coder cable.

SRVO-074 SERVO LDAL alarm(Group:%d Axis:%d)

Cause: LEDs in the pulse coder are disconnected.

Remedy: If this alarm occurs along with a SRVO-068 DTERR, SRVO-069 CRCERR, or SRVO-070 STBERR, disregard this alarm and refer to the other three alarm remedies. Replace the pulse coder and master the robot.

SRVO-075 WARN Pulse not established(G:%d A:%d)

Cause: The pulse coder does not know its own position yet.

Remedy: Jog the axis manually by more than one motor rotation.

SRVO-081 WARN EROFL alarm (Track enc:%d)

Cause: Line tracking encoder overflow due to high speed.

Remedy: None applicable.

SRVO-082 WARN DAL alarm(Track encoder:%d)

Cause: Line tracking pulse coder disconnected.

Remedy: Check for correct axis control PCB for line tracking and proper connections. Check line tracking pulse coder cables. Replace the SIF module on the axis control PCB. Replace the DSM module on the axis control PCB. Replace the pulse coder.

SRVO-083 WARN CKAL alarm (Track enc:%d)

Cause: The clock for the rotation counter in the line tracking pulse coder is abnormal.

Remedy: Refer to SRVO-061 remedy.

SRVO-084 WARN BZAL alarm (Track enc:%d)

Cause: The battery voltage for the line tracking pulse coder is zero volts.

Remedy: Refer to SRVO-062 remedy.

SRVO-085 WARN RCAL alarm (Track enc:%d)

Cause: The built-in rotation counter on the line tracking pulse coder is abnormal.

Remedy: Refer to SRVO-063 remedy.

SRVO-086 WARN PHAL alarm (Track enc:%d)

Cause: The relationship between the analog signals on the line tracking pulse coder are abnormal.

Remedy: Refer to SRVO-064 remedy.

SRVO-087 WARN BLAL alarm (Track enc:%d)

Cause: The line tracking pulse coder batteries are low.

Remedy: Refer to SRVO-065 remedy.

SRVO-088 WARN CSAL alarm (Track enc:%d)

Cause: The line tracking pulse coder ROM checksum data are abnormal.

Remedy: Refer to SRVO-066 remedy.

SRVO-089 WARN OHAL2 alarm (Track enc:%d)

Cause: The line tracking pulse coder overheated.

Remedy: Refer to SRVO-067 remedy.

SRVO-090 WARN DTERR alarm (Track enc:%d)

Cause: The axis control PCB sent the request signal, but did not receive serial data from the line tracking pulse coder.

Remedy: Refer to SRVO-068 remedy.

SRVO-091 WARN CRCERR alarm (Track enc:%d)

Cause: The serial data from the line tracking pulse coder changed during communication to the axis control PCB.

Remedy: Refer to SRVO-069 remedy.

SRVO-092 WARN STBERR alarm (Track enc:%d)

Cause: The communication stop and start bits for line tracking axis are abnormal.

Remedy: Refer to SRVO-070 remedy.

SRVO-093 WARN SPHAL alarm (Track enc:%d)

Cause: The feedback velocity exceeds specification for line tracking axis.

Remedy: Refer to SRVO-071 remedy.

SRVO-094 WARN PMAL alarm (Track enc:%d)

Cause: The interpolation circuits of the pulse coder are abnormal.

Remedy: Refer to SRVO-072 remedy.

SRVO-095 WARN CMAL alarm (Track enc:%d)

Cause: Line tracking encoder: Incorrect position data detected in the pulse coder, or abnormal pulse coder data caused by noise.

Remedy: Refer to SRVO-073 remedy.

SRVO-096 WARN LDAL alarm (Track enc:%d)

Cause: LEDs in the pulse coder are disconnected.

Remedy: Refer to SRVO-074 remedy.

SRVO-097 WARN Pulse not established(Enc:%d)

Cause: For line tracking encoder, the pulse coder does not its own position yet (due to improper installation).

Remedy: Refer to SRVO-075 remedy.

SRVO-101 SERVO Robot overtravel(Robot:%d)

Cause: A Robot overtravel limit switch, is pressed.

Remedy: Select the OT_RELEASE menus. Cursor to the axis that is overtraveled, OT_MINUS or OT_PLUS is TRUE, and press RELEASE. Press reset, wait for servo power to engage, and jog the robot off the overtravel switch. Check CRM11 connection on emergency stop control PCB.

SRVO-102 SERVO Hand broken (Robot:%d)

Cause: The hand broken (*HBK) robot input is asserted.

Remedy: If using *HBK input, determine the cause of the error and correct. If not, check the position of the *HBK jumper on the emergency stop control PCB; if on side A, *HBK is checked if on side B, *HBK is not checked *HBK originates on Main CPU PCB.

SRVO-103 SERVO Air pressure alarm(Rbt:%d)

Cause: The pneumatic pressure (PPABN) robot input is asserted.

Remedy: If using pneumatic pressure input clear source of fault, press reset. If pneumatic pressure is not used set \$PPABN_ENBL system variable to FALSE. PPABN originates on the Main CPU PCB.

SRVO-104 SERVO Welding electrode

Cause: No longer occurs Welding electrode of controller. This occurs only for the R-J2 Dual arm controller.

Remedy: Contact the FANUC Robotics hotline.

SRVO-111 SERVO Softfloat time out(G:%d)

Cause: Follow-up time is over when softfloat is ON.

Remedy: Make \$SFLT_FUPTIM larger.

SRVO-112 PAUSE Softfloat time out(G:%d)

Cause: Follow-up time is over when softfloat is ON.

Remedy: Make \$SFLT_FUPTIM larger.

SRVO-121 SERVO Excessive acc/dec time(G:%d)

Cause: Acceleration time is much longer for TurboMove case.

Remedy: Contact the FANUC Robotics hotline.

-
- SRVO-122 SERVO Bad last ang(internal)(G:%d)**
Cause: Last angle update request does not match current angle.
Remedy: Contact the FANUC Robotics hotline.
- SRVO-125 WARN Quick stop speed over (G:%d)**
Cause: Motion speed is too high to perform quick stop.
Remedy: Reduce the motion speed.
- SRVO-126 WARN Quick stop error (G:%d)**
Cause: A program was aborted during the servo quick stop process.
Remedy: Reset the system.
- SRVO-141 SERVO OHAL1(CNV) alarm (G:%d A:%d)**
Cause: Refer to SRVO-049.
Remedy: Refer to SRVO-049.
- SRVO-142 SERVO OHAL1(INV) alarm (G:%d A:%d)**
Cause: NOT used.
Remedy: N/A
- SRVO-143 SERVO PSFLAL(CNV) alarm (G:%d A:%d)**
Cause: Input power applied to amplifier is lost.
Remedy: Check the connections and cables of input power.
- SRVO-144 SERVO LVAL(INV) alarm (G:%d A:%d)**
Cause: Refer to SRVO-047.
Remedy: Refer to SRVO-047.
- SRVO-145 SERVO LVAL(CNV-DC) alarm(G:%d A:%d)**
Cause: Refer to SRVO-147.
Remedy: Refer to SRVO-147.
- SRVO-146 SERVO LVAL(INV-DC) alarm(G:%d A:%d)**
Cause: The DC voltage of the main circuit power supply is excessively low.
Remedy: Check each interphase voltage of the three-phase voltage (200 VAC) applied to the servo amplifier. If the applied voltage is found to be 170 VAC or less, check the input power supply voltage. Replace the servo amplifier.
- SRVO-147 SERVO LVAL(DCLK) alarm (G:%d A:%d)**
Cause: Back-up charge circuit for amplifier have trouble.
Remedy: Check the cables and connections between amplifier(CN1) and MCC. Check the fuse (F1,F3) in transformer. If using B-cabinet Replace the EMG Control printed circuit board. Replace the amplifier.
- SRVO-148 SERVO HCAL(CNV) alarm (G:%d A:%d)**
Cause: NOT used.
Remedy: N/A
- SRVO-149 SERVO HCAL(INV) alarm (G:%d A:%d)**
Cause: Refer to SRVO-045.
Remedy: Refer to SRVO-045.
- SRVO-150 SERVO FSAL(CNV) alarm (G:%d A:%d)**
Cause: Cooling fan for Control circuit stops.
Remedy: Check or Replace the fan.
- SRVO-151 SERVO FSAL(INV) alarm (G:%d A:%d)**
Cause: NOT used.
Remedy: N/A

SRVO-152 SERVO IPMAL(INV) alarm (G:%d A:%d)**Cause:** IPM module has trouble.**Remedy:** IPM might be overheated. Reset the emergency stop after approximately ten minutes. Disconnect the power lines from the terminals on the amplifier, and check the insulation of PE from U, V and W. If there are short-circuits, disconnect the motor connector power lines and check the insulation of PE from U, V and W.

1. Replace the motor if U, V and W short-circuit with PE.
2. Replace the power lines if U, V and W do not short-circuit with PE. Noise on the actual current(IR,IS) running in amplifier module might cause this alarm. Remove this noise such as with taking ground of sealed earth.
3. Replace the amplifier.

SRVO-153 SERVO CHGAL(CNV) alarm (G:%d A:%d)**Cause:** Charge of the main circuit could not finish within specified time.**Remedy:** DC link may short-circuit. Check the connections. Electric resistance to restrict charge current may be defective. Replace the wiring board.**SRVO-154 SERVO HVAL(CNV-DC) alarm (G:%d A:%d)****Cause:** Refer to SRVO-044.**Remedy:** Refer to SRVO-044.**SRVO-155 SERVO DCAL(CNV) alarm (G:%d A:%d)****Cause:** Refer to SRVO-043.**Remedy:** Refer to SRVO-043.**SRVO-160 SERVO Panel/External E-stop****Cause:** Either the operator panel emergency stop button was pressed, or the external emergency stop DI is input. This occurs only for R-J2 Mate.**Remedy:** Twist the operator panel emergency stop button clockwise to release it.
–If you are using external emergency stop, clear the source of the fault and press RESET.
–If not, check the wiring at EMGIN1, EMGIN2, and EMGINC on the EMG Control PCB. Check for 100 VAC input to the EMG Control PCB.**SRVO-161 SERVO Fence open or Deadman SW****Cause:** The teach pendant deadman switch is released or fence circuit is open.**Remedy:** Press teach pendant deadman switch or determine the cause of the fence open and press RESET.**SRVO-162 SERVO Deadman/Fence or Panel/External E-stop****Cause:** The deadman switch is released or fence circuit is open or the operator panel ESTOP button is pressed or external ESTOP signal is received.**Remedy:** Remove the cause then press RESET.**SRVO-163 SYSTEM DSM hardware mismatch****Cause:** Different DSM (Digital Servo Module) are mounted on controller.**Remedy:** Change DSM hardware to be same.**SRVO-164 SYSTEM DSM/Servo param mismatch****Cause:** DSM (Digital Servo Module) type is mismatched to servo parameter version.**Remedy:** Change current DSP IV (4) to DSP V (5) or initialize robot library again to load correct servo parameter file.**SRVO-165 SYSTEM Panel(SVON abnormal) E-stop****Cause:** The operator panel emergency stop push button is pressed and mis-wiring on SVON2 or EMG2 is detected.**Remedy:** Power off. Correct the wiring on SVON2 or EMG2. Power on. Twist the operator panel emergency stop push button clockwise to release. Press RESET.**SRVO-166 SYSTEM TP(SVON abnormal) E-stop****Cause:** The teach pendant emergency stop push button is pressed and miswiring on SVON2 or EMG2 is detected.**Remedy:** Power off. Correct the wiring on SVON2 or EMG2. Power on. Twist the teach pendant emergency stop push button clockwise to release. Press RESET.**SRVO-167 SYSTEM Deadman switch (SVON abnormal)****Cause:** The teach pendant deadman switch is released while the teach pendant is enabled. Miswiring on SVON2 or EMG2 is detected.**Remedy:** Power off. Correct the wiring on SVON2 or EMG2. Power on. Press teach pendant deadman switch. Press RESET.

-
- SRVO-168 SYSTEM External/SVON (SVON abnormal) E-stop**
Cause: Refer SRVO-007 or SRVO-019. Also miswiring on SVON2 or EMG2 is detected.
Remedy: Power off. Correct the wiring on SVON2 or EMG2. Power on. Refer SRVO-007 or SRVO-019.
- SRVO-171 WARN MotorSpd lim/DVC(G:%d A:%d)**
Cause: Motor can not rotate as fast as the calculated speed required for the current motion.
Remedy: This is just a notification. You do not have to do anything for this warning message.
- SRVO-172 WARN MotorSpd lim/DVC0(G:%d A:%d)**
Cause: Motor can not rotate as fast as the calculated speed required for the current motion.
Remedy: This is just a notification. You do not have to do anything for this warning message.
- SRVO-173 WARN MotorSpd lim/DVC1(G:%d A:%d)**
Cause: Motor can not rotate as fast as the calculated speed required for the current motion.
Remedy: This is just a notification. You do not have to do anything for this warning message.
- SRVO-174 WARN MotorAcc lim/DVC(G:%d A:%d)**
Cause: Motor can not accelerate as much as the calculated acceleration required to for the current motion.
Remedy: This is just a notification. You do not have to do anything for this warning message.
- SRVO-181 SERVO Mcmd input while estimating(G:%d)**
Cause: Robot was going to move while identifying the payload.
Remedy: Do not move the robot while identifying the payload. Press RESET.
- SRVO-192 SERVO Fence open/SVON input**
Cause: The fence circuit is open or SVON input circuit is open.
Remedy: Close the fence circuit or SVON input circuit, and then press RESET.
- SRVO-193 SERVO SVON input**
Cause: The SVON input circuit is open.
Remedy: Close the SVON input circuit and then press RESET.
- SRVO-194 SERVO Servo disconnect**
Cause: Servo is disconnected.
Remedy: Connect servo and then press RESET.
- SRVO-195 SERVO NTED/Servo disconnect**
Cause: Non Teacher Enabling Device is released or servo is disconnected.
Remedy: Press Non Teacher Enabling Device or connect servo, and then press RESET.
- SRVO-196 SYSTEM Fence open/SVON input (SVON abnormal)**
Cause: The fence circuit is open or the SVON input circuit is open and mis-wiring on SVON is detected.
Remedy: Power off. Correct the wiring on SVON. Close the fence circuit or SVON input circuit, and then press RESET.
- SRVO-197 SYSTEM SVON input (SVON abnormal)**
Cause: The SVON input circuit is open and mis-wiring on SVON is detected.
Remedy: Power off. Correct the wiring on SVON. Close the SVON input circuit, and then press RESET.
- SRVO-198 SYSTEM External E-stop (SVON abnormal)**
Cause: The external emergency stop push button is pressed and mis-wiring on SVON is detected.
Remedy: Power off. Correct the wiring on SVON. If using external emergency stop, clear the source of the fault and press RESET. If not, check the wiring at EMGIN1, EMGIN2, and EMGINC on the EMG control PCB. Check for 100 VAC input to the EMGM control PCB.
- SRVO-199 PAUSE Control Stop**
Cause: Control Stop is detected.
Remedy: After this alarm, fence open or SVON input alarm is detected. See the remedy of the next alarm.

SSPC Error Codes (ID = 69)

SSPC-002 STOP.G Occur dead lock condition

Cause: The priority of space is invalid.

Remedy: Set the priority valid.

SSPC-004 STOP.G CTV option not allowed

Cause: Space Check function is not compatible with Continuous Turn CTV option. The CTV motion option is not allowed.

Remedy: Remove CTV option or disable space check function.

SYST Error Codes (ID = 24)**SYST-001 PAUSE HOLD button is being pressed**

Cause: You attempted an operation while the hold button (input) is pressed.

Remedy: Clear the hold button (input), and try the same operation.

SYST-002 PAUSE HOLD is locked by program

Cause: The condition that the robot is being held is locked by the program, and it could not be cleared. If a HOLD statement is executed in a Karel program, the held condition can only be cleared by the same program using the UNHOLD statement/action, or by aborting the program. If a motion is attempted in such condition, this error message is displayed.

Remedy: Wait until the UNHOLD statement is executed by the karel program, or abort the KAREL program.

SYST-003 WARN TP is enabled

Cause: The attempted operation could not be done because the teach pendant is enabled.

Remedy: Disable the teach pendant, and try the same operation again.

SYST-004 WARN SOP is enabled

Cause: The attempted operation could not be done because the System Operator Panel is enabled.

Remedy: Turn the REMOTE switch on the SOP to REMOTE side, and try the same operation again.

SYST-005 WARN UOP is the master device

Cause: The attempted operation could not be done because the User Operator Panel is enabled.

Remedy: Turn the REMOTE switch to local (if the operation is attempted from the SOP), or set the \$RMT_MASTER system variable correctly. Refer to the *SYSTEM R-J2 Software Reference Manual*, Chapter 2 "System Variables", for more information on system variables.

SYST-006 WARN KCL is the master device

Cause: The attempted operation could not be done because KCL is the master device.

Remedy: Turn the REMOTE switch to local (if the operation is attempted from the SOP), or set the \$RMT_MASTER system variable correctly. Refer to the *SYSTEM R-J2 Software Reference Manual*, Chapter 2 "System Variables", for more information on system variables.

SYST-007 WARN NETWORK is the master device

Cause: The attempted operation could not be done because the NETWORK command processor is the master device.

Remedy: Turn the REMOTE switch to local (if the operation is attempted from the SOP), or set the \$RMT_MASTER system variable correctly. Refer to the *SYSTEM R-J2 Software Reference Manual*, Chapter 2 "System Variables", for more information on system variables.

SYST-008 WARN Nothing is the master device

Cause: The system variable \$RMT_MASTER is set to disable all devices. Therefore, no remote device can issue motion.

Remedy: Turn the REMOTE switch to local (if the operation is attempted from the SOP), or set the \$RMT_MASTER system variable correctly. Refer to the *SYSTEM R-J2 Software Reference Manual*, Chapter 2 "System Variables", for more information on system variables.

SYST-009 WARN Safety Fence open

Cause: The attempted operation could not be done because the safety fence is open.

Remedy: Close the safety fence, and try the same operation again.

SYST-010 WARN Max num task reached

Cause: The number of task has reached the maximum.

Remedy: Abort one of the running task.

SYST-011 WARN Failed to run task

Cause: The system has failed to run the program.

Remedy: Refer to the error cause code. Use MENU to display the Alarm Log screen.

SYST-012 WARN Not in remote

Cause: Remote condition is not satisfied.

Remedy: Turn the remote switch on.

SYST-013 WARN Invalid program number

Cause: The specified PNS number is not in the range of 1 to 9999.

Remedy: Specify correct program number.

SYST-014 WARN Program select failed

Cause: PNS operation has failed by some reason.

Remedy: Refer to the error cause code. Use MENU to display the Alarm Log screen.

SYST-015 WARN Robot Service Request failed

Cause: RSR operation has failed by some reason.

Remedy: Refer to the error cause code. Use MENU to display the Alarm Log screen.

SYST-016 WARN ENBL signal is off

Cause: ENBL signal in UOP is off.

Remedy: Set ENBL signal ON.

SYST-017 WARN Single step operation effective

Cause: Single step operation is effective.

Remedy: Disable single step switch.

SYST-018 WARN Continuing from different line

Cause: Attempt to continue program from different line from paused line.

Remedy: Respond YES or NO in the prompt box on at the teach pendant.

SYST-019 WARN Program not selected

Cause: Program has not been selected.

Remedy: Select a program from the program select menu on the teach pendant, or using PNS.

SYST-020 WARN Program not verified by PNS

Cause: Program specified by PNS is different from current selected program. This error occurs in R-J2 Mate only.

Remedy: Select a correct program from the program select menu on the teach pendant.

SYST-021 WARN System not ready, press reset

Cause: An error has been detected by the system.

Remedy: Press RESET to clear error condition.

SYST-022 WARN PNS not zero, cannot continue

Cause: Paused program cannot be continued if PNS input ports are not zero. This error occurs in R-J2 Mate only.

Remedy: Set all PNS input ports to OFF.

SYST-023 SYSTEM Teach Pendant communication error

Cause: A communication cable is broken.

Remedy: Check the teach pendant cable. Replace the cable if necessary.

SYST-024 WARN PNSTROBE is OFF. Cannot start exec

Cause: Because PNSTROBE is off, prod_start could not be processed.

Remedy: Set PNSTROBE input to ON.

SYST-025 WARN Teach Pendant is different type

Cause: The type of teach pendant being connected, is different from the one that was disconnected.

Remedy: Connect the same type of teach pendant as disconnected.

SYST-026 WARN System normal power up

Cause: System has executed normal power startup.

Remedy: This is just a notification. You do not have to do anything for this warning message.

SYST-027 PAUSE HOT start failed (Error:%d)

Cause: HOT start has failed for one of the following reasons:

1. Power failed during system start up.
2. Flash ROM module was changed.
3. A run-time error occurred.
4. System internal error 1.
5. System internal error 2.

Remedy: COLD start is selected automatically.

SYST-028 WARN (%s) Program timed out

Cause: \$PWR_HOT,\$PWR_SEMI program has been aborted by the system due to time out (40sec).

Remedy: Decrease program size so that it can be executed within the time out limit.

SYST-029 PAUSE Robot was connected (Group:%d)

Cause: The connect/isolate key was turn to the connect side.

Remedy: This is just a notification. You do not have to do anything for this warning message.

SYST-030 PAUSE Robot was isolated (Group:%d)

Cause: The connect/isolate key was turn to the isolate side.

Remedy: This is just a notification. You do not have to do anything for this warning message.

SYST-031 SYSTEM F-ROM parity

Cause: An error has occurred accessing FROM.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

SYST-032 WARN ENBL signal from UOP is lost

Cause: ENBL input signal from UOP is lost.

Remedy: Determine and correct the cause of loss of this signal.

SYST-033 WARN SFSPD signal from UOP is lost

Cause: SFSPD input signal from UOP is lost.

Remedy: Determine and correct the cause of loss of this signal.

SYST-034 WARN HOLD signal from SOP/UOP is lost

Cause: HOLD input signal from SOP/UOP is lost.

Remedy: Determine and correct the cause of loss of this signal.

SYST-035 WARN Low or No Battery Power in PSU.

Cause: Battery in PSU board is low in power.

Remedy: Replace the Old Battery with a new battery of same kind.

SYST-036 WARN Semi power failure recovery

Cause: System did semi-hot start.

Remedy: This is just a notification. You do not have to do anything for this warning message.

SYST-037 ABORT Key switch broken

Cause: Improper input from Key switch.

Remedy: Fix the CE Sign key switch.

SYST-038 PAUSE Operation mode T1 Selected

Cause: Operation mode T1 Selected.

Remedy: This is just a notification. You do not have to do anything for this warning message.

SYST-039 PAUSE Operation mode T2 Selected

Cause: Operation mode T2 Selected.

Remedy: This is just a notification. You do not have to do anything for this warning message.

SYST-040 PAUSE Operation mode AUTO Selected

Cause: Operation mode AUTO Selected.

Remedy: This is just a notification. You do not have to do anything for this warning message.

SYST-042 PAUSE DEADMAN defeated

Cause: The mode switch was changed from T1 or T2 mode to AUTO mode and the DEADMAN was already pressed. The DEADMAN must be released when switching to AUTO mode.

Remedy: Release the DEADMAN and press RESET.

SYST-043 PAUSE TP disabled in T1/T2 mode

Cause: The mode selector is in T1 or T2 and the TP ON/OFF switch is in the OFF position.

Remedy: Turn the TP ON/OFF switch to ON. Press RESET.

SYST-044 SYSTEM (Abnormal) TP disabled in T1/T2 mode

Cause: The mode selector is in T1 or T2 and the TP ON/OFF switch is in the OFF position and SVON is ON. This is an abnormal condition.

Remedy: Call your FANUC Robotics technical representative.

SYST-045 PAUSE TP enabled in AUTO mode

Cause: The mode selector is in AUTO and the TP ON/OFF switch is in the ON position.

Remedy: Turn the TP ON/OFF switch to OFF. Press RESET.

SYST-046 SYSTEM Control Reliable config mismatch

Cause: Either 1. Control Reliable hardware exists but the option has not been loaded, or 2. The Control Reliable option has been loaded but hardware is not available.

Remedy: If the option has not been loaded, load the Control Reliable option. If it has been loaded then this is a system without the Control Reliable hardware and the system must be totally reloaded WITHOUT the Control Reliable option.

SYST-047 WARN Continuing from distant position

Cause: Attempt to continue the program from a distant position from the stopped position.

Remedy: Select ABORT or CONTINUE in the prompt box displayed on the teach pendant.

TCPP Error Codes**TCPP-000 SWARN Unknown (TC00)****Cause:** A system error has occurred.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

TCPP-001 SWARN No global variables**Cause:** TCP Speed Prediction global variables are not loaded.**Remedy:** Perform a controlled start and initialize motion softparts.**TCPP-002 SWARN No mmr pointer****Cause:** This is an internal system error.**Remedy:** Perform a cold start on the controller.**TCPP-003 SWARN No mir pointer****Cause:** This is an internal system error.**Remedy:** Perform a cold start on the controller.**TCPP-004 SWARN No sysvar pointer****Cause:** This is an internal system error.**Remedy:** Perform a cold start on the controller.**TCPP-005 SWARN No tcppir pointer****Cause:** This is an internal system error.**Remedy:** Perform a controlled start and initialize the motion softparts.**TCPP-006 SWARN Error allocating memory****Cause:** A failure occurred while allocating memory.**Remedy:** Check amount of memory being used by system.**TCPP-007 SWARN Error making TCPP mailbox****Cause:** This is an internal system error.**Remedy:** Perform a cold start on the controller.**TCPP-008 SWARN Error making TCPP spmktsk****Cause:** This is an internal system error.**Remedy:** Perform a cold start on the controller.**TCPP-009 SWARN Pnts too close before line:%d^5****Cause:** The program positions prior to this line are too close together to allow proper speed prediction to be performed. It is also most likely that the robot can not perform at the programmed speed either.**Remedy:** The positions being executed prior to the one specified should be moved further apart, or the programmed speed should be decreased.**TCPP-010 SWARN Error writing PLAN mailbox****Cause:** This is an internal system error.**Remedy:** Perform a cold start on the controller.**TCPP-011 SWARN Pred time skips first motion****Cause:** The first motion segment time (time between first two taught points) is shorter than the prediction equipment delay time as specified by \$TCPPIR.\$TCDELAY.**Remedy:** Extend time of first motion segment (increase distance or decrease speed) to allow TCP Speed Prediction to begin during the first motion segment.**TCPP-012 SWARN Invalid TCPP filter type****Cause:** The motion was commanded using an invalid motion filter type.**Remedy:** Check program setup to ensure proper program header data, motion types, and position data.**TCPP-013 SWARN Invalid TCPP start pos****Cause:** The start position is not valid.**Remedy:** Check the values for the current and previous positions.**TCPP-014 SWARN Invalid TCPP destination pos****Cause:** The destination position is not valid.**Remedy:** Check the values for the current and previous positions.

TCPP-015 SWARN Invalid time into segment

Cause: The time computed for the current segment exceeds the segment length.

Remedy: Check the value for the current position data.

TCPP-016 SWARN No cfseg data

Cause: The Cartesian Filter data is not valid.

Remedy: Check to make sure that the Cartesian Filter option has been properly loaded and initialized.

TCPP-017 SWARN Modone computation error

Cause: The internal Motion Done computation generated an error.

Remedy: No action is necessary, but the TCP Speed value will not be valid for several motions.

TCPP-018 SWARN Begin Error Mode at line:%d^5

Cause: The program positions beginning with this line have caused an error condition which makes further speed prediction invalid. The speed prediction task has therefore entered an error handling mode within which it discontinues speed prediction but provides the actual robot speed as an output reference with an effective equipment delay time of 0. (That is, with \$TCPPIR.\$TCDELAY = 0.)

Remedy: Check the error log to find speed limit errors which may be causing this TCPP error to occur. Check the program beginning with this line to determine if the robot can not function as programmed. Modify the program to avoid all such speed limit errors. In cases which list speed limit errors for the wrist axes, the error can often be eliminated by changing the motion command from using mm/sec speed commands to using deg/sec speed commands which better control motion with large wrist orientation changes.

TCPP-019 SWARN Speed Ovrld Mode at line:%d^5

Cause: A speed override occurred beginning with this line causing an error condition which makes further speed prediction invalid. The speed prediction task has therefore entered an error handling mode within which it discontinues speed prediction but provides the actual robot speed as an output reference.

Remedy: Normal prediction will resume automatically once the speed override has propagated through the system. No further action is required. The user should avoid this during production program execution and should be aware that it may invalidate live application testing since neither the robot nor the speed prediction task are performing as programmed.

TPIF Error Codes (ID = 9)**TPIF-001 WARN Mnemonic editor error (%s^1)**

Cause: Illegal case occurred on software.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

TPIF-002 WARN Operating system error (%s^1)

Cause: Illegal case occurred on software.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

TPIF-003 WARN Window I/O error (%s^1)

Cause: Illegal case occurred on software.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

TPIF-004 WARN Memory write error

Cause: Illegal case occurred on software.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

TPIF-005 WARN Program is not selected

Cause: The program was not selected yet, when the program was displayed at the TEACH screen.

Remedy: Select a program in the SELECT screen.

TPIF-006 WARN SELECT is not taught

Cause: This taught statement needed the SELECT statement before the current line.

Remedy: Teach the SELECT statement before the current line.

TPIF-007 WARN Robot is not calibrated

Cause: The robot has not been calibrated properly.

Remedy: Calibrate the robot properly.

TPIF-008 WARN Memory protect violation

Cause: The program is write protected.

Remedy: Release protection of the program on the SELECT screen.

TPIF-009 WARN Cancel delete by application

Cause: Program is protected.

Remedy: Release protection of the program on the SELECT screen then delete the program.

TPIF-010 WARN Cancel enter by application

Cause: Program is protected.

Remedy: Try edit after release protection by application.

TPIF-011 WARN Item is not found

Cause: Item is not found below this line.

Remedy: Try another item or close search function.

TPIF-012 WARN Kinematics solution is invalid

Cause: Cannot translate position data.

Remedy: Check the configuration of robot and \$MNUTOOL/\$MNUFRAM system variables.

TPIF-013 WARN Other program is running

Cause: You cannot select a program when another program is running or paused.

Remedy: Select a program after aborting the program which is currently running or paused.

TPIF-014 WARN Teach pendant is disabled

Cause: You cannot edit a program when the teach pendant is disabled.

Remedy: First enable the teach pendant, then edit the program.

TPIF-015 WARN Bad position register index

Cause: Specified a invalid index of position register.

Remedy: Check the index of position register.

TPIF-016 WARN Memory access failed (%s^1)

Cause: Illegal case occurred on software.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

TPIF-017 WARN Memory read failed

Cause: Illegal case occurred on software.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

TPIF-018 WARN Unspecified index value

Cause: Specified index value is invalid.

Remedy: Check specified index value.

TPIF-019 WARN This item cannot be replaced

Cause: This item cannot be replaced.

Remedy: Try another item or close replace function.

TPIF-020 NONE Mnaction search error

Cause: Illegal case occurred on software.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

TPIF-021 NONE Mnteach software error

Cause: Illegal case occurred in software.

Remedy: Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

TPIF-023 WARN WJNT and RTCP are not compatible

Cause: Wjnt and RTCP are not compatible.

Remedy: Remove Wjnt or RTCP before adding the other.

-
- TPIF-030 WARN Program name is NULL**
Cause: Program name is not set properly.
Remedy: Set a proper program name.
- TPIF-031 WARN Remove num from start of Program name**
Cause: Start of program name is numeric.
Remedy: Remove numeric value from beginning of program name.
- TPIF-032 WARN Remove space from Program name**
Cause: Space is included in program name.
Remedy: Remove space from program name.
- TPIF-033 WARN Remove comma from Program name**
Cause: Comma is included in program name.
Remedy: Remove comma from program name.
- TPIF-034 WARN Remove dot from Program name**
Cause: Dot is included in program name.
Remedy: Remove dot from program name.
- TPIF-035 WARN Remove minus from Program name**
Cause: Minus is included in program name.
Remedy: Remove minus from program name.
- TPIF-036 WARN Not enough memory**
Cause: There is not enough memory available.
Remedy: Delete unused program.
- TPIF-037 WARN Program must be selected by TP**
Cause: Only the Teach Pendant default program can be edited on the CRT.
Remedy: Select the program on the Teach Pendant before editing on the CRT.
- TPIF-038 WARN Invalid char in program name**
Cause: Invalid character in program name.
Remedy: Remove invalid character from program name.
- TPIF-040 WARN Label already exists**
Cause: Same label No. already exists.
Remedy: Change to different label No.
- TPIF-041 WARN MNUTOOLNUM number is invalid**
Cause: Specified MNUTOOLNUM number is invalid.
Remedy: Check system variable \$MNUTOOLNUM.
- TPIF-042 WARN MNUFRAMENUM number is invalid**
Cause: Specified MNUFRAMNUM number is invalid.
Remedy: Check system variable \$MNUFRAMNUM.
- TPIF-043 WARN External change is valid**
Cause: You cannot change the robot (group), because the function that select robot by external DI is valid.
Remedy: Set system variable \$MULTI_ROBO.CHANGE_SDI to ZERO.
- TPIF-044 WARN Program is unsuitable for robot**
Cause: The group mask of program differs from selected robot (group).
Remedy: Check selected robot (group) or check program attributes group mask.
- TPIF-045 WARN Pallet number is over max**
Cause: Cannot teach more than 16 Palletizing instructions in one program.
Remedy: Teach another program.
- TPIF-046 WARN Motion option is over max**
Cause: Too many motion options for default motion.
Remedy: Decrease motion options for default motion.
- TPIF-047 WARN Invalid program is selected**
Cause: Program type is wrong.
Remedy: Select TPE program.

TPIF-048 WARN Running program is not found

Cause: There is currently no program running that can be monitored.
Remedy: Run program before attempting to monitor.

TPIF-049 WARN Port number is invalid

Cause: Port is not set for outside device.
Remedy: Set port for outside device.

TPIF-050 WARN Macro does not exist

Cause: A program is not assigned to this macro command.
Remedy: Assign a program to this macro command.

TPIF-051 WARN Program has been selected by PNS

Cause: When a program has been selected by PNS, you cannot select program from SELECT screen.
Remedy: Turn off the PNSTROBE signal.

TPIF-052 WARN FWD/BWD is disabled

Cause: When the Disabled FWD function has been selected, you cannot execute the program from the teach pendant.
Remedy: Select the Disabled FWD in the function menu, then you can release from the Disable FWD.

TPIF-053 WARN Not editing background program

Cause: The program has not been selected by BACKGROUND editing.
Remedy: Select the BACKGROUND program in the SELECT screen.

TPIF-054 WARN Could not end editing

Cause: 1. There is not enough memory.
2. The background program is invalid.
Remedy: 1. Delete unnecessary programs.
2. Confirm the background program.

TPIF-055 WARN Could not recovery original program

Cause: Failed recovering original program which has been selected by the BACKGROUND.
Remedy: End editing by the END_EDIT of [EDCMD] again before executing the original program which has been selected by the BACKGROUND.

TPIF-056 WARN This program is used by the CRT

Cause: The program of BACKGROUND cannot be selected by the CRT and TP at the same time.
Remedy: End editing by the END_EDIT of [EDCMD] at the CRT.

TPIF-057 WARN This program is used by the TP

Cause: The program of BACKGROUND cannot be selected by the CRT and TP at the same time.
Remedy: End editing by the END_EDIT of [EDCMD] at the TP.

TPIF-060 WARN Can't record on cartesian (G:%d)

Cause: This current position is in singularity.
Remedy: You can record this position on joint type only by selecting the function key.

TPIF-061 WARN Group[%s] has not recorded

Cause: This position data has not been changed to displayed groups because you selected the function key which did not record the position, when checking in singularity.
Remedy: Check this recorded position again before execution.

TPIF-062 WARN AND operator was replaced to OR

Cause: All AND operators on this line were replaced with OR operators.
Remedy: You cannot mix AND and OR operator on a the same line. Verify that all logical operators on this line are the same before execution.

TPIF-063 WARN OR operator was replaced to AND

Cause: All OR operator on this line were replaced by AND operators. You cannot mix AND OR operator on a the same line.
Remedy: Verify all logical operators on this line before execution.

TPIF-064 WARN Too many AND/OR operator(Max.4)

Cause: Too many AND/OR operators (Max.4 on a single line).
Remedy: Teach the logical operation on another line.

-
- TPIF-065 WARN Arithmetic operator was unified to +- or */**
Cause: Arithmetic operator on this line was changed to + – or * /. Cannot mix arithmetic + and – operators with * and / operators on the same line.
Remedy: Verify all arithmetic operators on this line before execution.
- TPIF-066 WARN Too many arithmetic operator(Max.5)**
Cause: Too many arithmetic operators (Max.5 on a single line).
Remedy: Teach the arithmetic operation on another line.
- TPIF-070 WARN Cannot teach the instruction**
Cause: Cannot teach the instruction.
Remedy: Check the sub type of the program.
- TPIF-071 WARN Cannot change sub type**
Cause: Cannot change sub type.
Remedy: Check sub type of the program.
- TPIF-072 WARN Cannot change motion group**
Cause: Cannot change motion group.
Remedy: Check sub type of the program.
- TPIF-090 WARN This program has motion group**
Cause: The program specified in \$PWR_HOT, \$PWR_SEMI and \$PWR_NORMAL must not have motion group.
Remedy: Set * to all motion group in program detail screen on the teach pendant.
- TPIF-091 WARN PREG access error**
Cause: An error occurred when accessing a position register.
Remedy: Refer to the error cause code on the ALARM log screen.
- TPIF-092 WARN Value %d expected %s**
Cause: The value_array that was passed to a built-in was incorrectly specified. The error line shows the index into value_array where the error occurred and the type expected by the built-in.
Remedy: Make sure the value_array specifies the correct names for the variables and that the types expected are correct.
- TPIF-093 WARN USER menu must be selected**
Cause: A KAREL program called a user interface built-in which required the USER menu to be displayed on the teach pendant or CRT.
Remedy: Use FORCE_SPMENU(tp_panel, SPI_TPUSER, 1) before calling the user interface built-in on the teach pendant.
Use FORCE_SPMENU crt_panel, SPI_TPUSER, 1) before calling the user interface built-in on the CRT.
- TPIF-094 WARN USER2 menu must be selected**
Cause: A KAREL program called a user interface built-in which required the USER2 menu to be displayed on the teach pendant or CRT.
Remedy: Use FORCE_SPMENU(tp_panel, SPI_TPUSER2, 1) before calling the user interface built-in on the teach pendant.
Use FORCE_SPMENU crt_panel, SPI_TPUSER2, 1) before calling the user interface built-in on the CRT.
- TPIF-095 WARN Execution history table error**
Cause: Software internal error.
Remedy: Perform a controlled start (it isn't necessary to re-set the new item).
- TPIF-097 WARN Can't display running task's history**
Cause: The execution history of the executing program cannot be displayed.
Remedy: Use this screen when the program is paused or aborted.
- TPIF-098 WARN %s was not run**
Cause: The program of \$PWR_HOT, \$PWR_SEMI or \$PWR_NORMAL is not executed.
Remedy: Refer to the error cause code. Use the Alarm Log screen.
- TPIF-099 WARN This program is being edited**
Cause: The program specified in \$PWR_HOT, \$PWR_SEMI and \$PWR_NORMAL is not executed, when the program is in editing.
Remedy: Select the other program.

TPIF-100 WARN No vacant table space**Cause:** Illegal case occurred on software.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

TPIF-101 WARN No such menu**Cause:** Illegal case occurred on software.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

TPIF-102 WARN E.STOP is asserted**Cause:** FWD execution is selected while, E-STOP is asserted.**Remedy:** Turn the E-STOP off. Then select FWD execution.**TPIF-103 WARN Dead man is released****Cause:** FWD execution is selected while, DEADMAN switch is released.**Remedy:** Press and hold down the DEADMAN, then select FWD execution**TPIF-104 WARN Teach Pendant is disabled****Cause:** FWD execution is selected while, TP is disabled.**Remedy:** Enable the teach pendant. Then select FWD execution.**TPIF-105 WARN Program is not selected****Cause:** FWD execution is requested without selection of program.**Remedy:** Select a program for execution. Then select FWD execution**TPIF-106 WARN Program is already running****Cause:** FWD execution is requested when program is running.**Remedy:** Abort the running program before requesting FWD execution.**TPIF-107 WARN FWD/BWD is disabled****Cause:** When the Disabled FWD function has been selected, you cannot execute the program from the teach pendant.**Remedy:** Select the Disabled FWD in the function menu, then you can release from the Disable FWD**TPIF-108 WARN Form error, line %d, item %d****Cause:** The Form Manager detected an error on the specified line with the specified item.**Remedy:** Refer to the cause code on the ALARM log screen for the actual error.**TPIF-109 WARN %v not specified correctly****Cause:** The Form Manager detected an error when displaying a %v item.**Remedy:** To specify the %v enumeration type in a form dictionary, use lower case v followed by the dictionary element which specifies the program name and variable name of the variable which contains the display values. For example: "Enum Type: " "-%6v(enum_fkey)" \$-,enum_fkey "TPEX" &new_line "CHOICE_ARRAY" In the above example, CHOICE_ARRAY is a KAREL string array variable in program TPEX which contains the enumeration choices. The enumeration choices are displayed in a subwindow.**TPIF-110 WARN Screen used by other device****Cause:** The screen you are attempting to use on the teach pendant is currently displayed on the CRT. Or the screen you are attempting to use on the CRT is currently displayed on TP**Remedy:** Exit from the screen on the other device.**TPIF-116 WARN System variable error: %s****Cause:** System variable name is invalid.**Remedy:** Check the spelling and format of the name.**TPIF-117 WARN Cannot backup to device: %s****Cause:** The default device is not valid for backup.**Remedy:** Select a valid device and try again.

TPIF-118 WARN File error for %s**Cause:** File error.**Remedy:** Perform a cold start:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot. If the error is not cleared, document the events that led to the error and call your FANUC Robotics technical representative.

TPIF-119 WARN File compression failed**Cause:** Failed creating compressed file.**Remedy:** Check backup device.**TPIF-120 WARN Device failure****Cause:** Device failure.**Remedy:** Check device and try again.**TPIF-121 WARN Invalid copy. Use MOVE key.****Cause:** Cannot COPY a file on a Memory device to the same Memory device.**Remedy:** Use the MOVE key and try again.**TPIF-128 WARN Verify logic of pasted line(s).****Cause:** The reverse motion copy function does not support the following motion option instructions:

- Application command
- Skip, Quick Skip
- Incremental
- Continuous turn
- Time before/Time after

Remedy: Check the above motion instructions and modify the copied statement correctly.**TPIF-129 WARN Group motion inst. is pasted****Cause:** The group motion instruction is copied. The reverse motion copy function does not support group motion instructions.**Remedy:** Check the motion instruction and modify the copied statement correctly.**TPIF-132 WARN Can't recover this operation****Cause:** Because the data for UNDO cannot be saved, this operation cannot be recovered by the UNDO function.**Remedy:** Check the cause code. If the memory is full, please delete the program, or disable the UNDO function.**TPIF-133 WARN Can't recover this command****Cause:** Palletizing command and compliance control cannot be recovered by the UNDO function.**Remedy:**

VARs Error Codes (ID = 16)

VARs-001 WARN Corrupt variable file

Cause: An error has occurred trying to read specified file. This file is corrupt or the media is bad.

Remedy: Try a different file.

VARs-002 WARN Open Error on File

Cause: The variable file does not exist on the device, director or media.

Remedy: Place correct media in drive or select the proper device/directory and try again.

VARs-003 WARN %s array length updated

Cause: A variable being loaded from a variable file exists in memory. The array length reflects what was in the variable file.

Remedy: This is just a notification. You do not have to do anything for this warning message.

VARs-004 WARN %s memory not updated

Cause: A variable being loaded from a variable file exists in memory. The variable file data cannot be loaded.

Remedy: Clear the program and load the variables first before loading program.

VARs-005 WARN %s PC array length ignored

Cause: A variable being loaded from a variable file exists in memory. The array length reflects what was in the variable file.

Remedy: This is just a notification. You do not have to do anything for this warning message.

VARs-006 WARN Unknown Variable Name

Cause: Referenced variable does not exist.

Remedy: Load PC file or VR file to create the variable.

VARs-007 WARN Unknown Type Code

Cause: Referenced type code does not exist.

Remedy: Load PC file or VR file to create the type.

VARs-008 WARN Type Name not found

Cause: Referenced type name does not exist.

Remedy: Load PC file or VR file to create the named type.

VARs-009 WARN SV Load at CTRL Start Only

Cause: A variable load has been requested while controller is capable of motion.

Remedy: Create an error condition such as E-stop and load of variables is allowed.

VARs-010 WARN Variable/field write-protected

Cause: The variable or field you are trying to access is write protected.

Remedy: This variable is not to be changed by customer for safety or other reasons. If you are trying to change \$SCR variables just change \$PARAM_GROUP and cold start.

VARs-011 WARN No data defined for program

Cause: Referenced program name does not have variables.

Remedy: Load PC file or VR file to create the named program.

VARs-012 WARN Create var - %s failed

Cause: Named variable could not be created.

Remedy: Refer to the error cause code. Use the Alarm Log screen to display the cause code.

VARs-013 WARN Variable Already Exists

Cause: Referenced variable already exist in memory.

Remedy: This is just a notification. You do not have to do anything for this warning message.

VARs-014 WARN Create type - %s failed

Cause: Named type could not be create.

Remedy: Refer to the error cause code. Use the Alarm Log screen to display the cause code.

VARs-015 WARN Too many vars/nodes/programs

Cause: The limit of variables types, programs or nodes has been reached.

Remedy: You must delete some programs or reorganize programs to make more room.

VARs-016 WARN Axis configuration mismatch

Cause: The variables you are trying to load are were created on a controller with a different axis configuration.

Remedy: These variable cannot be used on this controller.

VAR5-017 WARN Sysvar version mismatch

Cause: The system variable file you are attempting to load is not compatible with the loaded software version.
Remedy: You must use the default system variable file supplied with your version of software.

VAR5-018 WARN Compatible Type Already Exists

Cause: Referenced type already exists in memory.
Remedy: This is just a notification. You do not have to do anything for this warning message.

VAR5-019 WARN Rename target exists

Cause: You are attempting to rename a program to a program which already exists.
Remedy: Use a different program name or delete the program and variables from existing program.

VAR5-020 WARN [%s]%s not fnd

Cause: Referenced variable is not found in the system.
Remedy: Load PC file or VR file to create the variable.

VAR5-021 WARN Memory allocation failure

Cause: There is no more permanent memory available in the system.
Remedy: You must delete unneeded programs, dictionaries or variables to make room.

VAR5-022 WARN Duplicate creation TYPE mismatch

Cause: Variable that is being created already exists but is of a different type than what you are attempting to load/create.
Remedy: Delete existing variable before creating it as a different type.

VAR5-023 WARN Array len creation mismatch

Cause: Variable that is being created already exists but has different dimensions than what you are attempting to load/create.
Remedy: Delete existing variable before creating it with conflicting dimensions.

VAR5-024 WARN Bad variable or register index

Cause: You are attempting to use an invalid index into an array or path.
Remedy: Use a valid index.

VAR5-025 WARN Vision reference error

Cause: Do not have vision hardware on this system so cannot load vision variables.
Remedy: Load these variables on an appropriate system.

VAR5-026 WARN File sequence error

Cause: The file which has been loaded is:
– Not a variable file
– A file on bad media
– A file not compatible with your current software.
Remedy: Try a different file or convert the current file to an updated version.

VAR5-027 WARN Variable used by other program

Cause: Variable is used by another program.
Remedy: Delete other program which references these variables.

VAR5-028 WARN Value out of range

Cause: Value that you entered is not a valid value. It is either too big or too small.
Remedy: Consult your *SYSTEM R-J2 Controller Software Reference Manual* for valid values for the variable you are changing.

VAR5-029 WARN Requires PROGRAM password

Cause: The operation that you are attempting is password protected.
Remedy: You must go to the password setup screen and enter the PROGRAM password.

VAR5-030 WARN Requires SETUP password

Cause: The operation that you are attempting is password protected.
Remedy: You must go to the password setup screen and enter the SETUP password.

VAR5-031 WARN Requires INSTALL password

Cause: The operation that you are attempting is password protected.
Remedy: You must go to the password setup screen and enter the INSTALL password.

VAR5-032 WARN Variable size too big

- Cause:** The variable you are loading is larger than 65,535 bytes or has an array element larger than 32,767 bytes.
Remedy: Make the array size smaller or use a path data type for large arrayed variables Maximum path length is 2,007 Maximum node size is 32,767.

VAR5-033 WARN Maximum path length exceeded

- Cause:** A path can only contain 2,007 nodes.
Remedy: You must break up the large path into smaller paths

VAR5-034 WARN Variable cannot be accessed

- Cause:** The CMOS variable you tried to delete was created at controlled start, or a variable in the program you were trying to access had another read write operation in progress. This could be because a KAREL program, Network or KCL was adding deleting or doing a node operation when access was attempted.
Remedy: Delete the variable in the start mode in which it was created. Attempt the operation again when no other variable accesses are in progress.

VAR5-036 WARN CMOS memory is corrupt

- Cause:** CMOS memory has been destroyed.
Remedy: Controller initial start must be performed .

VAR5-037 WARN Position register is locked

- Cause:** Position register is locked by program operation.
Remedy: Wait until program is finished.

VAR5-038 WARN Cannot change CMOS/DRAM type

- Cause:** An existing variable is being created in a different memory area (CMOS vs DRAM).
Remedy: Delete the variable or change the memory type to be used.

VAR5-039 WARN Data set created

- Cause:** Permanent memory was successfully allocated.
Remedy: This is just a notification. You do not have to do anything for this message.

VAR5-040 WARN Cannot load at CONTROL START 2

- Cause:** Variables may not be properly created if loaded at this time.
Remedy: Load variables at COLD start or at CONTROLLED START 1 before save image operation.

VAR5-041 WARN Invalid Node Number

- Cause:** Path insertion or delete of a node occurred with node number which exceeded the number of nodes in a path.
Remedy: Perform operation with a valid node number.

VAR5-042 WARN TEMP type invalid for CMOS create

- Cause:** The type definition for the variable being created is in temporary DRAM memory. This means variable cannot be remembered after power off.
Remedy: The program with the type definition for the variable you are creating must be loaded at controlled start. This implies the type definition is image.

VAR5-043 WARN Variable memory pool is invalid

- Cause:** The memory pool for this variable does not exist on the controller. An auxiliary board has probably been removed or replaced.
Remedy: Put the old board in back into the controller. If this board is not available then an INITIAL START is required .

VAR5-053 WARN Input data pointer invalid

- Cause:** An invalid pointer was sent to the controller from a PC.
Remedy: Check all pointers being sent from the PC using RPC calls.

WNDW Error Codes (ID = 18)**WNDW-001 WARN Invalid screen name format**

Cause: Format of screen name in DEF_SCREEN, ACT_SCREEN, or ATT_WINDOW_S call is invalid.

Remedy: Screen names must be 1-4 alpha characters Supply a valid screen name.

WNDW-002 WARN Invalid window name format

Cause: Format of window name in ATT_WINDOW_D, ATT_WINDOW_S, or DET_WINDOW call or an OPEN statement is invalid.

Remedy: Window names must be 1-4 alpha characters Supply a valid window name.

WNDW-003 WARN Invalid keybd. name format

Cause: Invalid display device name in DEF_SCREEN or ATT_WINDOW_D call.

Remedy: Use TP for teach pendant screen or CRT for KCL screen.

WNDW-004 WARN Invalid disp dev name format

Cause: Invalid format of display device name in DEF_SCREEN or ATT_WINDOW_D call.

Remedy: Device names must be 1-4 alpha characters Supply a valid device name.

WNDW-005 WARN Bad number of rows

Cause: Invalid n_rows in DEF_WINDOW call.

Remedy: n_rows value must be 1 to 50 Correct the value.

WNDW-006 WARN Bad number of cols

Cause: Invalid n_cols in DEF_WINDOW call.

Remedy: n_cols value must be 1 to 132 Correct the value.

WNDW-007 WARN Bad row number

Cause: Invalid value of row parameter in ATT_WINDOW_S, AT_WINDOW_D, or SET_CURSOR call.

Remedy: For ATWINDOW_S or AT_WINDOW_D calls, row must be in the range 1-(display_device_size-window_size+1) Correct the row parameter value For SET_CURSOR calls, the value must be in the range 1-50.

WNDW-008 WARN Bad col number

Cause: Invalid value of col parameter in ATT_WINDOW_S, AT_WINDOW_D, or SET_CURSOR call.

Remedy: For ATWINDOW_S or AT_WINDOW_D calls, col must be in the range 1-(display_device_size-window_size+1) Correct the col parameter value For SET_CURSOR calls, the value must be in the range 1-132.

WNDW-011 WARN Unk. disp dev name

Cause: Unknown display device name in DEF_SCREEN or ATT_WINDOW_D call.

Remedy: Use TP for teach pendant screen or CRT for KCL screen.

WNDW-012 WARN Unk k/b dev name

Cause: Keyboard device specified in a PUSH_KEY_RD or POP_KEY_RD call or OPEN statement is invalid.

Remedy: Use 'TP' for teach pendant keys or 'CRT' for KCL keyboard.

WNDW-013 WARN Duplicate screen name

Cause: Screen name specified in DEF_SCREEN call is already defined.

Remedy: If the screen is system defined, it cannot be redefined. If the existing screen definition is not being changed, this may not be a problem. Otherwise, it may be necessary to cold-start the controller to delete the existing definition:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot.

WNDW-014 WARN Duplicate window name

Cause: Window name specified in DEF_WINDOW call is already defined.

Remedy: If the window is a system defined window, it cannot be redefined. If the existing window definition is not being changed, this may not be a problem. Otherwise, it may be necessary to cold-start the controller to delete the existing definition:

1. Turn off the robot.
2. On the teach pendant, press and hold the SHIFT and RESET keys.
3. While still pressing the SHIFT and RESET keys, turn on the robot.

WNDW-015 WARN Unknown screen name

Cause: The screen name specified in a ATT_WINDOW_S, DET_WINDOW, or ACT_WINDOW call is not defined.
Remedy: Use the name of a defined screen.

WNDW-016 WARN Unknown window name

Cause: The window name specified in a ATT_WINDOW_S, ATT_WINDOW_D, or DET_WINDOW call or an OPEN statement is not defined.
Remedy: Use the name of a defined window.

WNDW-017 WARN Window already attchd to scrn

Cause: The window name specified in a ATT_WINDOW_S, ATT_WINDOW_D is ready attached to the specified screen.
Remedy: If the present attach is acceptable, this may not be a problem. Otherwise, it will be necessary to call DET_WINDOW.

WNDW-018 WARN Invalid file name string

Cause: The file name in an OPEN statement begins WD: or KB: but is not a valid format.
Remedy: The following forms are valid:
 WD:wnam
 KB:knam
 WD:wnam/knam
 KB:knam/wnam where wnam and knam are 1-4 alpha numeric characters. Correct the format of the file name.

WNDW-019 WARN Write to file w/o window

Cause: A write was issued to a file opened to a keyboard (KB:knam), but not a window.
Remedy: Either change the OPEN to specify a window or do not WRITE to the file.

WNDW-020 WARN Bad buffer length on read

Cause: The buffer_size parameter in a INI_DYN_DISI INI_DYN_DISR, INI_DYN_DISS or READ_KB call is invalid.
Remedy: Specify a value in the range 10-128 for INI_DYN_DISx calls; for READ_KB calls, specify a range of 1-128, unless the accept mask is zero, when a buffer_size value of zero is permitted.

WNDW-021 WARN Invalid timeout value

Cause: The time_out value in a READ_KB call is Invalid.
Remedy: The value must be less than 65535000. Use a valid value.

WNDW-022 WARN Zero term char mask in read

Cause: The terminate character mask in a keyboard read is zero.
Remedy: Use a non-zero value for the terminate character mask.

WNDW-023 WARN Initial data too long

Cause: The init_data value is longer than the buffer size parameter.
Remedy: Either increase the value of buffer_size or specify a shorter string for init_value.

WNDW-024 WARN Attempt to read with no kb

Cause: A READ_KB call or READ statement was executed on a file that was OPENed to a window but not a keyboard.
Remedy: Either modify the OPEN FILE statement to specify a keyboard or do not use the file in a READ_KB call or READ statement.

WNDW-025 WARN Echo window for read not act

Cause: A READ_KB call was executed where the specified file is opened to a window that is not attached to the active screen and the terminate mask included the no_window bit.
Remedy: This may be a normal result if the user intended READ_KB requests to fail if the required window is not displayed. Otherwise, either modify the terminate mask or use ATT_WINDOW_D or ATT_WINDOW_S to attach the required window.

WNDW-026 WARN Read for same keys/kbd active

Cause: A READ_KB call was executed where the the keyboard for the specified file currently has another READ_KB call or READ statement that accepts some of the same classes of keys and the terminate mask included the kbd_busy bit.
Remedy: This may be a normal result if the user intended READ_KB requests to fail if the keyboard is in use. Otherwise, modify the terminate mask, modify the accept mask of this or the conflicting read, or use a PUSH_KEY_RD call to suspend conflicting reads.

WNDW-027 WARN Too many pushes active

Cause: The maximum depth of key read PUSH operations has been exceeded.

Remedy: Check for situations in which a PUSH_KEY may be executed and no POP_KEY is executed.

WNDW-028 WARN Mis-match on push/pop seq

Cause: This indicates that the pop_index specified in a POP_KEY_RD call is not the expected value, indicating that call are being made out of order.

Remedy: Check the logic in use of PUSH_KEY_RD and POP_KEY_RD to ensure that the pop_index values are being supplied in the correct order. If more than one task is issuing PUSH_KEY_RD and POP_KEY_RD calls, extra care is required.

WNDW-030 WARN Invalid time

Cause: The interval parameter in a INI_DYN_DISI, INI_DYN_DISR, or INI_DYN_DISS call is invalid.

Remedy: This must be in the range 1-32767 (ms).

WNDW-032 WARN No match on var disp cncl

Cause: There is no currently active dynamic display for variable and window specified in a CNC_DYN_DISI, CNC_DYN_DISR, or CNC_DYN_DISS call.

Remedy: Check the variable and window names. Also check logic to see that dynamic display had been started and not already cancelled.

WNDW-033 WARN Field width invalid

Cause: The field_width parameter in a call to one of the INI_DYN_DIS builtin routines is invalid.

Remedy: Value must be in the range of 0-255.

**B CRT/KB SETUP
AND OPERATION**

B CRT/KB SETUP AND OPERATION

Topics In This Appendix		Page
CRT/KB Setup	You connect a remote CRT/KB to any RS-232-C port on the controller. You set up this port according to the requirements of your CRT/KB.	B-2
CRT/KB Menus	The contents of the menus on the CRT/KB match the menus on the teach pendant except that the CRT/KB does not include any menus that involve robot motion	B-2
CRT/KB Keys	The correspondence between CRT/KB and teach pendant keys is shown in this section	B-2

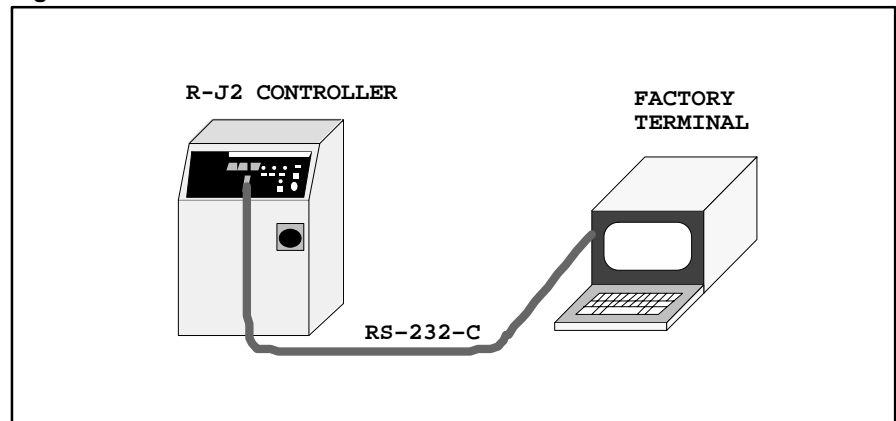
The cathode ray tube/keyboard (CRT/KB) is an optional user interface device you can use, in addition to the teach pendant to display teach pendant screens and perform robot operations. In general, you can perform any robot operation from the CRT/KB except operations that involve moving the robot, such as jogging and test cycle.

The CRT/KB can be external to the controller, or remote. You can use the following types of remote CRT/KBs:

- Factory terminal
- DEC VT-220 terminal
- IBM PC-compatible computer with VT-220 terminal emulation software

Figure B-1 shows an example of a remote CRT/KB.

Figure B-1. Built-in and Remote CRT/KBs



This appendix describes how to set up and operate the CRT/KB.

B.1 CRT/KB SETUP

You connect a remote CRT/KB to any RS-232-C port on the controller. You set up this port according to the requirements of your CRT/KB. The factory terminal is listed in Table B-1. Refer to the manufacturing specifications of any other type of remote CRT/KB for port setup information.

Table B-1. Port Settings for the Factory Terminal

Speed	Parity Bit	Stop Bit	Timeout Value
9600 baud	None	1 bit	0 sec

Refer to Chapter 9 for information on setting up ports.

B.2 CRT/KB MENUS

The contents of the menus on the CRT/KB match the menus on the teach pendant except that the CRT/KB does not display

- Any menus that involve robot motion.
- The SETUP Touch I/O screen
- The SETUP Touch Frame screen

B.3 CRT/KB KEYS

The correspondence between CRT/KB and teach pendant keys is shown in Table B-2. You cannot jog the robot from the CRT/KB, so no jog keys exist. Numeric keys on the CRT/KB correspond directly to numeric keys on the teach pendant. Alphabetic keys on the CRT/KB are used for direct alphabetic entry.

Table B-2. Correspondence Between Teach Pendant and CRT/KB Keys

Teach Pendant Key	CRT/KB Key
F1, F2, F3, F4, F5	F1, F2, F3, F4, F5
Arrow keys	Cursor keys
SHIFT + UP arrow key (page up)	F7
SHIFT + DOWN arrow key (page down)	F8
ITEM	F6
FCTN	F9
MENUS	F10
—	DO key for KCL*

* For DEC VT-220 terminals only

C BOOTROM OPERATIONS

C

BOOTROM OPERATIONS

Topics In This Appendix		Page
Startup Methods	You can start up the robot and controller using one of the following start methods:	C-2
	• INIT Start	C-2
	• Controlled Start (START CTRL)	C-3
	• Controlled 2 Start (START CTRL2)	C-6
	• Cold Start (START COLD)	C-7
	• Semi Hot Start	C-9
	• Re-Init Start (CMOSINIT)	C-10
BootROM Utilities	You can access the following utilities:	C-12
	• Extended Boot Monitor (EMON>) Utilities	C-15
	• Diagnostic Utilities	C-16
	• INSTALL Utilities	C-18
	• Flash ROM Utilities	C-20
	• Memory Card Utilities	C-21

The BootROM is a device you can use to turn on the robot using different start methods and to use specific system utilities. The BootROM hardware consists of the BootROM EPROM chip, located on the Main CPU PCB.

C.1 STARTUP METHODS

BootROM provides the following startup methods:

- INIT start
- Controlled start/Controlled 2 start
- Cold start - *standard start method* (approximately 30 seconds)
- Semi Hot start - *standard start method* (approximately 15 seconds)
- Re-Init start



WARNING

DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.

C.1.1 INIT Start

An initialized (INIT) start occurs automatically as the first phase of software installation.



CAUTION

Do not use init start to start the controller. An INIT start erases all information stored in the saved memory pools. CMOS must be manually cleared before you can perform an INIT start. Instead, **use re-init start** (CMOSINIT) to start the controller. Refer to Section C.1.6.

C.1.2 Controlled Start (START CTRL)

A *controlled start (START CTRL)* turns on power to the robot and controller and allows you to do the following:

- Set robot motion parameters
- Install options and updates
- Load or set system variables

CAUTION

The items that appear on the controlled start screen control how the robot and controller operate. Do not set these items unless you are certain of their effect, otherwise, you could disrupt the normal operation of the robot and controller.

Table C-1. Controlled Start Options

ITEM	DESCRIPTION
Motion System Variable Setup	<ul style="list-style-type: none"> ● Initialize System Variable Reruns the robot library setup program for the selected motion group. ● Add/Delete Group Allows you to add and delete motion groups. ● Extended Axis Setup/Init Allows you to set up and initialize extended axes. ● Initialize Motion Softpart Initializes any softparts attached to motion that have not yet been initialized. ● Display System Setup Status Displays the current robot library and whether it is initialized.
Program Initialization	Allows you to set the maximum number of tasks, number of registers, and number of position registers in the controller.
Motion Development	<ul style="list-style-type: none"> ● Disable Digital Servo Program Start When FALSE, activates the servo system; TRUE does not activate the servo system. ● Start Motion Test Task For FANUC Robotics internal use only. ● Enable CMOS Servo Code For FANUC Robotics internal use only. ● Use CMOS Servo Code For FANUC Robotics internal use only.

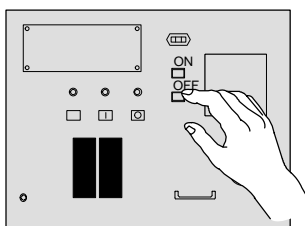
Table C-1. (Cont'd) Controlled Start Options

ITEM	DESCRIPTION
<p>Exit</p>	<p>Displays the Configuration screen. Refer to the <i>FANUC Robotics SYSTEM R-J2 Controller Application-Specific Software Installation Manual</i> for more information on setting these items. Press FCTN and select START (COLD) when you are done.</p> <ul style="list-style-type: none"> • MENU key <ul style="list-style-type: none"> – S/W INSTALL Used to install software. – S/W VERSION Displays the STATUS VERSION ID screen. – Variables Displays the SYSTEM Variables screen. – File Displays the FILE screen. – ALARM Displays the ALARM screen. – Port Init Displays the FILE Port Init screen – MEMORY Displays the STATUS MEMORY screen • FCTN key <ul style="list-style-type: none"> – START (COLD) Performs a COLD START. Refer to Section C.1.4. – START (CTRL2) Performs a CTRL2 START. Refer to Section C.1.3. – SAVE Saves current data. – PRINT SCREEN Prints the current screen to a serial printer or, if a PC is connected to the P3 port, to a file called TPSCRN.LS. – PRINT CONFIG Prints softpart configuration information to a serial printer or, if a PC is connected to the P3 port, to a file called CONFIG.LS. – UNSIM ALL I/O Unsimulates all I/O settings. • Application Setup <ul style="list-style-type: none"> – Performs application-specific setup and initialization. • Install Option <ul style="list-style-type: none"> – Used to install standard software options. • Install Update <ul style="list-style-type: none"> – Used to install update software.

Use Procedure C-1 to perform a controlled start.

Procedure C-1 Performing a Controlled Start

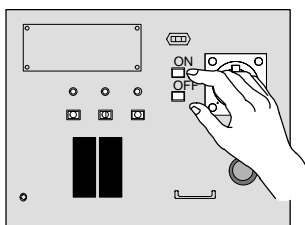
Condition ■ All personnel and unnecessary equipment are out of the workcell.



i-size controller operator box

⚠ WARNING
DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.

- 1 **If the controller is turned on**, turn it off.
- 2 On the teach pendant, press and hold the PREV and NEXT keys.
- 3 While still pressing PREV and NEXT on the teach pendant, press the ON button on the operator box or operator panel.



i-size controller operator box

BMON>

- 4 After the **BMON>** prompt appears on the teach pendant screen, release the PREV and NEXT keys.

BMON> CTRL

- 5 Press F2, CTRL, and press ENTER.

BMON> START

- 6 Press F5, START, and press ENTER. This begins the controlled start. You will see a screen similar to the following.

```
Controlled Start Initialization

1 MOTION SYSVAR SETUP
2 PROGRAM INIT
3 MOTION DEVELOPMENT
4 EXIT

Press enter or number key to select.
```

- 7 Select the kind of setup or initialization you want to perform and continue as directed by the prompts on the screen and the information specific to your installation.

⚠ CAUTION

Never turn off the robot after a START (COLD) or START (CTRL2) has been selected (when the message, "System save in progress" is displayed on the teach pendant). Otherwise, you will corrupt the controller and all software will have to be reloaded.

- 8 **When you are finished** with the Controlled Start Initialization:
 - a Press 4, EXIT.
 - b **If you want to set configuration items**, refer to the *FANUC Robotics SYSTEM R-J2 Software Installation Manual* for more information on setting these items. Press FCTN and select START (COLD) when you are finished.
 - c **If you want to operate the robot**, perform a cold start. Press FCTN and select START (COLD).
 - d **If you want to load system variable files**, press MENUS, select FILE and load .SV or .VR files. Refer to Chapter 9, "Program and File Manipulation."
 - e **If you want to load teach pendant programs at this time**, you must perform a controlled 2 start. You cannot load teach pendant files at a Controlled Start. Refer to Section C.1.2.

C.1.3 Controlled 2 Start (START CTRL2)

A *controlled 2 start (START CTRL2)* updates memory and enables you to load teach pendant programs. It is also the mode the controller must be in to perform a full controller backup. Refer to Section 9.4. Controlled 2 start is the second phase of a controlled start.

NOTE You cannot load system variable files (.SV), or install options or updates during a CTRL2 START.

Use Procedure C-2 to perform a controlled 2 start.

Procedure C-2 Performing a CTRL2 Start

Condition ■ All personnel and unnecessary equipment are out of the workcell.

⚠ WARNING
DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.

- Step**
- 1 Perform a controlled start. Refer to Procedure C-1 , Steps 2 through 6.
 - 2 On the teach pendant, you will see a screen similar to the following.
 - 3 Select 4, EXIT and press ENTER.
 - 4 Press F4, YES.
 - 5 Press FCTN.

⚠ CAUTION
The CTRL2 start takes a few minutes to finish. Do not turn off the controller until the CTRL2 start has completed. Otherwise, you will lose the software loaded on your controller and will have to reload it. The CTRL2 start is finished when the FCTN menu disappears and you can display it again by pressing the FCTN key.

- 6 Select START (CTRL2) and press ENTER. The CTRL2 start will be performed immediately. When it is finished, you will see a title line on the screen similar to the following.

HANDLING CONFIG	CONTROL 2 START MENUS
1 / 20	

- 7 When the CTRL2 start has completed, press FCTN.

NOTE After a CTRL2 START is performed, item 2 on the FCTN menu will be blank.

- 8 Load teach pendant programs** if necessary. Press MENUS and select FILE.
- 9** When you have finished, select START (COLD). A cold start will be performed.
- 10 To restore files**, refer to Section 9.4.

C.1.4 Cold Start (START COLD)

A *cold start (START COLD)* is the standard method for turning on power to the robot and controller. A cold start does the following:

- Initializes changes to system variables
- Initializes changes to I/O setup
- Displays the UTILITIES Hints screen

A cold start will be complete in approximately 30 seconds.

Use Procedure C-3 to perform a cold start.

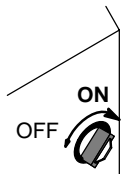
Procedure C-3 Performing a Cold Start

- Condition**
- All personnel and unnecessary equipment are out of the workcell.



WARNING

DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.



CIRCUIT
BREAKER

- Step**
- 1** Visually inspect the robot, controller, workcell, and the surrounding area. During the inspection make sure all safeguards are in place and the work envelope is clear of personnel.
 - 2** Turn the power disconnect circuit breaker on the operator box or operator panel to ON.
 - 3** On the teach pendant, press and hold the PREV and NEXT keys.

BMON>

4 After the **BMON>** prompt appears on the teach pendant screen, release the PREV and NEXT keys.

BMON> COLD

5 Press F1, COLD, and press ENTER.

BMON> START

6 Press F5, START, and press ENTER.

- **On the operator panel or operator box**, the ON button will be illuminated, indicating robot power is on.
- **On the teach pendant screen**, you will see a screen similar to the following.

```

UTILITIES Hints                               JOINT 10 %

MENUS - Provides a list of all menus.
FCTNS - Provides helpful functions.
SELECT, EDIT, DATA, POSN, I/O, STAT -
    Displays menus with those names.
Function keys with [ ] show more choices.
F1 [ TYPE ] function key lists related
    screens with a menu.

[ TYPE ]                                     HELP

```

For North American HandlingTool, you will see a screen similar to the following.

```

UTILITIES Hints                               JOINT 10 %

HandlingTool (TM)

V4.40-1

Copyright 1998, FANUC Robotics
North America, Inc.
All Rights Reserved

[ TYPE ]                                     HELP

```

C.1.5 Semi Hot Start

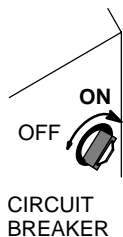
Semi hot start is one of the standard methods for turning on power to the robot and controller without using BootROM. Semi hot start is active when the system variable \$SEMIPOWERFL is set to TRUE. By default, \$SEMIPOWERFL is set to FALSE. You perform a semi hot start by pressing the ON button. The screen that was displayed before power was turned off is displayed.

When \$SEMIPOWERFL is set to TRUE, after the first cold start the controller is put into semi hot start mode automatically. This means that the next time you turn on the controller (by pressing the ON button), a semi hot start will be performed. A semi hot start will be complete in approximately half the time of a cold start.

If a program was running at the time power was turned off, the program will be paused when power is turned on. Use the standard methods of resuming a paused program. If \$SEMIPOWERFL is set to FALSE and power is turned off while a program is running, when power is turned on, the program will be ABORTED and cannot be resumed.

The semi hot start procedure is the same as the procedure for turning on the robot. Use Procedure C-4 to perform a semi hot start.

Procedure C-4 Performing a Semi Hot Start

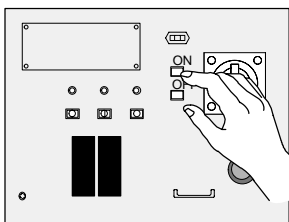


Condition

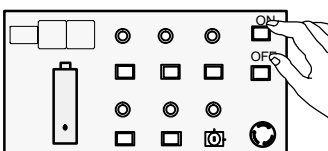
- All personnel and unnecessary equipment are out of the workcell.

Step

- 1 Visually inspect the robot, controller, workcell, and the surrounding area. During the inspection make sure all safeguards are in place and the work envelope is clear of personnel.
- 2 Turn the power disconnect circuit breaker on the operator panel or operator box to ON.



i-size controller operator box



B-size controller operator panel



WARNING

DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.

- 3 Press the ON button on the operator panel.

- **On the operator panel or operator box**, the ON button will be illuminated, indicating robot power is on.
- **On the teach pendant screen**, you will see the screen displayed when the robot was last turned off.

C.1.6 Re-Init Start (CMOSINIT)

A re-init start will cause all CMOS resident softparts to be reloaded. This is useful when some portion of CMOS memory has become fragmented, or accidentally overwritten.

- **Performing a re-init start** – Use this when the TPP, PERM, or IMAGE memory pools have become fragmented, or accidentally overwritten. A re-init start can also be performed if the robot library information was installed incorrectly.

The file, CMOSINIT.CF, located on the disk labeled AP1, executes all of the proper steps needed to perform a re-init start. A re-init start

- Clears CMOS memory
- Loads the TPE memory pool configuration
- INIT starts the controller

Use Procedure C-5 to perform a re-init start.

Procedure C-5 Performing a Re-Init Start using CMOSINIT

- | | |
|------------------|---|
| Condition | <ul style="list-style-type: none"> ■ A portion of CMOS memory is overwritten or otherwise corrupted, IMAGE memory has been depleted, or you have installed the wrong robot library during software installation. ■ You have obtained a PS-100 or PS-110 disk drive, memory card interface, or a PC compatible computer. ■ You have backed up all your teach pendant programs and other files you want to save. Refer to Chapter 9. ■ You have the R-J2 software disks that you want to install. ■ The PS-100 disk drive, memory card interface, or PC compatible computer is connected to the controller and is turned on. (Refer to Section 9.1.2.) |
| Step | <ol style="list-style-type: none"> 1 If the controller is turned on, turn it off. 2 Insert the disk labeled AP1 in the disk drive. 3 Press and hold the PREV and NEXT keys on the teach pendant. |

- 4 Turn on the controller. You will see a screen similar to the following.

```

*** BOOT MONITOR for R-J2 CONTROLLER ***
Version 4.22                                01-JAN-9x

F-ROM/D-RAM/C-MOS : 6.0/8.0/1.0 MB
TP Version         : I
Current TIME       : 01-JAN-199x 22:52:53

Slot  ID      FC      OP
0     9B      0       0    R-J2 Main CPU
1     6A      0       0    AB/Ether I/F optional
D     6A      0       0    MCARD I/F optional

BMON>
COLD   CTRL   INIT   NOLOAD  START  >

```

- 5 Turn on the disk drive.
- 6 Press NEXT, >, until F2, INSTALL is displayed.
- 7 Press F2, INSTALL, and then press ENTER.
- 8 Press NEXT, >, until F2, RUN is displayed.
- 9 Press F2, RUN.
- 10 Press F3, CMOSINIT and then press ENTER. You will see a screen similar to the following.

```

Slot  ID      FC      OP
0     9B      0       0    R-J2 Main CPU
1     6A      0       0    AB/Ether I/F
D     6A      0       0    MCARD I/F

BMON> INSTALL
INSTALL> RUN CMOSINIT

Run file CMOSINIT.CF

Are you sure ? (Y=1/N=0) :

```

- 11 **If you do not want to continue**, press 0. The BMON> prompt will be displayed.
- If you are ready to continue**, press 1.
- 12 To set up your robot model and your application, refer to the *FANUC Robotics SYSTEM R-J2 Software Installation Manual*.
- 13 Re-install all options and updates. Refer to the *FANUC Robotics SYSTEM R-J2 Software Installation Manual*.

C.2 BOOTROM UTILITIES

You can perform operations from the BootROM prompt, `BMON>`. Table C-2 lists and describes the utilities you can use from the `BMON>` prompt.

- The `EMON` utility provides access to BootROM extended monitor utilities. Refer to Section C.2.1.
- The `DIAG` utility provides access to BootROM diagnostic utilities. Refer to Section C.2.2.
- The `INSTALL` utility provides access to installation utilities. Refer to Section C.2.3.
- The `FROM` utility provides access to Flash ROM memory utilities. Refer to Section C.2.4

Use Procedure C-6 to access BootROM and use BootROM utilities.

Table C-2. BootROM Utilities

UTILITY	DESCRIPTION
COLD	Performs a cold start. Refer to Section C.1.4.
CTRL	Performs a controlled start. Refer to Section C.1.2.
INIT	Performs an initialized start. CAUTION An initialized start should only be used for a full software load. Controller memory is altered, and software that is currently on the controller is lost and cannot be recovered. Use Re-init start instead. Refer to Section C.1.6.
NOLOAD	Prevents automatic loading of the system memory area from Flash ROM to D-RAM. IMAGE is always loaded.
START	When the controller is powered up, START begins whatever kind of start (COLD, CTRL, or INIT) has been chosen. Semi Hot start cannot be selected.

Table C-2. (Cont'd) BootROM Utilities

UTILITY	DESCRIPTION
CLEAR	<p data-bbox="505 407 1159 434">Clears C-MOS RAM, D-RAM, Flash ROM or MCARD memory.</p> <p data-bbox="505 459 1398 514">CAUTION This can destroy the contents of C-MOS RAM memory, D-RAM memory, Flash ROM memory and MCARD. This includes all programs and files.</p> <p data-bbox="505 548 1365 630">Clear CMOS Clears the entire CMOS memory with zeros. You are prompted to confirm the execution of the file; answer YES to confirm, NO to cancel.</p> <p data-bbox="505 640 1360 722">Clear DRAM Clears the entire DRAM system code area with zeros. You are prompted to confirm the execution of the file; answer YES to confirm, NO to cancel.</p> <p data-bbox="505 732 1321 787">Clear DRAM FFFFFFFF Clears the entire CMOS memory with FFFFFFFF. For FANUC Robotics use only.</p> <p data-bbox="505 798 1146 825">Clear FROM ALL Clears the entire FROM memory.</p> <p data-bbox="505 835 1349 917">Clear FROM CHIP <i>n</i> Clears the entire FROM memory where <i>n</i> is 1 for the first 2 megabyte area; 2 for the second 2 megabyte area, and so forth.</p> <p data-bbox="505 928 1377 1010">Clear FROM Block (SYST, IMAG, SYSR) Clears the FROM memory save block. You are prompted to confirm the execution of the file; answer YES to confirm, NO to cancel.</p> <p data-bbox="505 1020 1373 1102">Clear MCARD Clears the memory card with zeros. You are prompted to confirm the execution of the file; answer YES to confirm, NO to cancel.</p>
EMON	Provides access to the extended monitor utilities. Refer to Section C.2.1.
CRT	Causes all screen information to be displayed on the CRT/KB. Pressing TP SELECT key causes the display to toggle between the teach pendant and the RS-232-C port. The SELECT key toggles between the teach pendant screen and the CRT device.
DIAG	Provides access to the diagnostics utilities. Refer to Section C.2.2.
INSTALL	Provides access to BMON software installation utilities. Refer to Section C.2.3.
FROM	Provides access to Flash ROM utilities. Refer to Section C.2.4.
MCARD	Provides access to memory card software installation utilities. Refer to Section C.2.5.

Procedure C-6 Using BootROM Utilities

⚠ WARNING
DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.

Condition Step

- The controller is turned off.
- 1 Press and hold the PREV and NEXT keys on the teach pendant, then press the ON button.

The boot monitor prompt, **BMON>**, is displayed. You will see a screen similar to the following.

```

*** BOOT MONITOR for R-J2 CONTROLLER ***
Version 4.22                               01-JAN-199x

F-ROM/D-RAM/C-MOS : 6.0/8.0/1.0 MB
TP Version        : I
Current TIME      : 01-JAN-199x 22:52:53

Slot  ID      FC      OP
0     9B      0       0     R-J2 Main CPU
1     6A      0       0     AB/Ether I/F
D     6A      0       0     MCARD I/F
BMON>
COLD  CTRL    INIT    NOLOAD  START  >
    
```

optional
 optional

- 2 To display more commands, press NEXT, >.

⚠ CAUTION
 The INIT utility deletes some of the current software in the controller. Do not use the INIT utility unless you want to reload the software; otherwise, a loss of data will occur.

- 3 To execute a command, press the appropriate function key and press ENTER.
- 4 To enter data manually,
 - Press the down arrow to display the alphabet from beginning to end
 - Press the up arrow to display the alphabet from end to beginning
 - Enter a character by pressing the right arrow

C.2.1 Extended Boot Monitor (EMON>) Utilities

The extended boot monitor utility (EMON) is used to load and clear memory on sub CPUs such as Vision (VISN), or Ethernet/RIO (ENAB) CPU.

You can access extended boot monitor utilities from BootROM. Table C-3 lists and describes the extended boot monitor utilities. Use Procedure C-7 to access extended boot monitor utilities.

Table C-3. BootROM Extended Monitor Utilities

UTILITY	DESCRIPTION
EXIT	Returns to BMON>
ECLEAR FROM mid ECLEAR CMOS mid	Clears the Flash ROM or C-MOS on other boards in the backplane. <i>mid</i> = Module ID such as VISN, ENAB, or RISC. CAUTION This can destroy the contents of C-MOS RAM memory, D-RAM memory, Flash ROM memory and MCARD. This includes all programs and files.
ELOAD FILE_NAME mid	Loads software from the specified file to hardware specified by <i>mid</i> . <i>mid</i> = Module ID for modules such as VISN, ENAB, or RISC.

Procedure C-7 Using EMON> Utilities

- Condition** ■ The BMON> prompt is displayed. Refer to Procedure C-6 .
- Step**
- 1 On the teach pendant, press NEXT, >, until F3, EMON, is displayed.
 - 2 Press F3, EMON.
 - 3 Press ENTER. You will see a screen similar to the following.

```
BMON >
EMON>
EXIT                ECLEAR    ELOAD    >
```

- 4 To display more utilities, press NEXT, >.

CAUTION

The EMON utilities invalidate the current vision software in the controller. Do not use these utilities unless you want to reload the software; otherwise, a loss of data will occur.

- 5 To execute a utility, press the appropriate function key and press ENTER.
- 6 To exit EMON, type ENTER at the EMON> prompt and press ENTER. The BMON> prompt will be displayed.

C.2.2 Diagnostic Utilities

The boot monitor diagnostic utilities are used to maintain and diagnose controller setup and hardware.

You can access diagnostic utilities from BootROM. Table C-4 lists and describes the diagnostic utilities. Use Procedure C-8 to access diagnostic utilities.

Table C-4. BootROM Diagnostic Utilities

UTILITY	DESCRIPTION
EXIT	Returns to BMON>.
GOFF	Used to shut down any GFS/KFLOPPY process running on a remote PC device.
DB hex_addr	Displays the memory, in byte/word/long word format, after you provide the hexadecimal starting memory location.
DW hex_addr	
DL hex_addr	
SHOW CONF	Shows the configuration of FROM as: <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> NAME: FRCONF: DRCONF: TPESIZE: Allocated Blocks: Free Memory: </div>
SHOW TIME	Shows the controller time.
SHOW MODULE	Shows the hardware modules, which also are displayed on the first BMON screen.
SHOW ENETADDR	Shows the Ethernet address. This will display an error message unless the Ethernet software is loaded on the Ethernet board and the Ethernet board is installed. Refer to the <i>SYSTEM R-J2 Ethernet Controller Backup and Restore – FTP Setup and Operations Manual</i> for more information.
SHOW MEMORY	Shows the BMON memory usage. For FANUC Robotics use only.
SET PORTn SPEED value SET PORTn DEVICE value	Sets the speed of the specified port n (n=2,3,4): 19200, 9600, 4800, 2400 Sets the device of the specified port n (n=2,3,4): Greco (FLPY:), CRT
SET TIME	Sets controller time.
SET ENETADDR	Sets the Ethernet hardware address. This will display an error message unless the Ethernet software is loaded on the Ethernet board and the Ethernet board is installed. Refer to the <i>SYSTEM R-J2 Ethernet Controller Backup and Restore – FTP Setup and Operations Manual</i> for more information.
MB hex_addr value*	Allows you to modify the memory, by byte/word/long format.
MW hex_addr value*	
ML hex_addr value*	
FRCONF value*	For FANUC Robotics internal use only.
DRCONF value*	For FANUC Robotics internal use only.

UTILITY	DESCRIPTION
BALLOC name value*	For FANUC Robotics internal use only.
SYSNAME 'char string'*	For FANUC Robotics internal use only.
CONFIG *	For FANUC Robotics internal use only.
TPESIZE value*	For FANUC Robotics internal use only.
CHGPATH 'char string'*	For FANUC Robotics internal use only.
TEST CMOS * TEST DRAM * TEST FROM *	Tests the memory by writing and reading. You are prompted to confirm the execution of the file; answer YES to confirm, NO to cancel. This displays the addresses as they are tested.

⚠ CAUTION

All functions marked with an asterisk (*) can modify controller memory. If they are used incorrectly they will corrupt your controller.

Procedure C-8 Using DIAG> Utilities

- Condition** ■ The BMON> prompt is displayed. Refer to Procedure C-6 .
- Step**
- 1 On the teach pendant, press NEXT, >, until F2, DIAG, is displayed.
 - 2 Press F2, DIAG.
 - 3 Press ENTER. You will see a screen similar to the following.

```

DIAG>
EXIT      GOFF      DB      DW      DL      >

```

- 4 **To display more utilities**, press NEXT, >.
- 5 **To execute a utility**, press the appropriate function key and press ENTER.
- 6 **To exit EMON**, type ENTER at the DIAG> prompt and press ENTER. The BMON> prompt will be displayed.

C.2.3 INSTALL Utilities

You can access the INSTALL utilities from BootROM to install software. Table C-5 lists the items you can install using the INSTALL utilities. Use Procedure C-9 to access INSTALL utilities. The BMON INSTALL utilities use three file devices at once. First BMON looks for memory card (MC:), then ENET (if loaded and installed on the Ethernet board), then P2: (FLPY:).



CAUTION

All functions marked with an asterisk (*) can modify controller memory. If they are used incorrectly they will corrupt your controller.

Table C-5. INSTALL Utilities

UTILITY	DESCRIPTION
EXIT	Exits to BMON>.
SHD	Displays the current directory.
CHD 'char string'	Change directory to the specified character string.
DIR	Performs a directory of the files on the first device found: NOTE If the CHD function has not been executed, the directory is performed in the following order: MCARD, ENET, then FLPY. Otherwise, the directory is performed on the device that was specified in the CHD command.
LOAD <i>filename</i> *	Loads the file into DRAM or C-MOS RAM. <i>filename</i> is the name of the file to be loaded. You will be prompted to confirm the load; answer YES to confirm, NO to cancel.
FS LOAD <i>filename</i> *	Loads the file directly to FROM. For FANUC Robotics use only.
ENET (optional)	This will display an error message unless the Ethernet software is loaded on the Ethernet board and the Ethernet board is installed. Refer to the <i>SYSTEM R-J2 Ethernet Controller Backup and Restore</i> manual for more information. This starts the BOOTP client looking for the BOOTP server. NOTE You must have already performed the SET ENETADDR function to set the Ethernet address before you can execute ENET.
RUN <i>filename</i>	Runs the specified command file. The following command files (.CF) can be run: <ul style="list-style-type: none"> • AUTOLOAD – Does not automatically set a standard TPE size for your application. • UPDATE – Found only on a maintenance update disk. This is used to load software that cannot be loaded at a controlled start. • CMOSINIT – Clears all the memory pools on the D-RAM and CMOS RAM devices. It also causes these memory pools to be reloaded from the F-ROM device. • FROMINIT – Clears all the memory pools on the F-ROM, D-RAM, and CMOS devices. It causes them to be reloaded from the ENET (if installed), floppy, or MCARD device. It will load ENAB, RISC, or VISN CPUs. • AUTO – Similar to FROMINIT but does not load non-main processors. You are prompted to confirm the execution of the file; answer YES to confirm, NO to cancel.
FTEST <i>filename</i> 'prompt string'	For auto loading, tests for the presence of the specified file, <i>filename</i> . If the file is not there, it prompts the user with the 'prompt string.' For FANUC Robotics use only.
RESTORE*	Executes the restore.cf file to restore a backup set of software. You are prompted to confirm the execution of the file; answer YES to confirm, NO to cancel.
ORD LOAD* ORD EDIT*	For FANUC Robotics use only.

Procedure C-9 Using INSTALL Utilities

- Condition** ■ All personnel and unnecessary equipment are out of the workcell.

⚠ WARNING
DO NOT turn on the robot if you discover any problems or potential hazards. Report them immediately. Turning on a robot that does not pass inspection could result in serious injury.

- The BMON> prompt is displayed. Refer to Procedure C-6 .

Step

BMON> INSTALL

- 1 Press NEXT, >, until F2, INSTALL is displayed.
- 2 Press F2, INSTALL and press ENTER. You will see a screen similar to the following.

```
INSTALL>
EXIT          SHD          CHD          DIR          >
```

INSTALL>

- 3 At the INSTALL> prompt, press NEXT, >, until F2, RUN is displayed.

INSTALL> RUN

- 4 Press F2, RUN. You will see a screen similar to the following.

```
INSTALL> RUN
AUTOLOAD UPDATE CMOSINIT FROMINIT AUTO
```

Refer to Table C-5 for a description of each of these files.

5 To reload software,

- a Press F4, FROMINIT and press ENTER. You will see a screen similar to the following.

```
Slot  ID      FC      OP      R-J2 Main CPU
0      9B      0       0
1      6A      0       0      AB/Ether I/F
D      6A      0       0      MCARD I/F

BMON> INSTALL
INSTALL> FROMINIT

RUN FILE FROMINIT.CF

Are you sure ? (Y=1/N=0) :
```

- b **If you do not want to continue**, press 0. The BMON> prompt will be displayed.

If you are ready to continue, press 1.

- c Re-install all options or updates.

C.2.4 Flash ROM Utilities

You can access the Flash ROM (F-ROM or FROM disk) utilities from BootROM. Table C-6 lists the FROM items you can use. Use Procedure C-10 to access the FROM utilities.

Table C-6. Flash ROM Items

ITEM	DESCRIPTION
EXIT	Exits to BMON>.
FRDB addr*	Displays Flash ROM memory address in byte/word/long word format. FROM addresses start at relative 0.
FRDW addr*	
FRDL addr*	
FRSAVE name start_addr size*	Saves SYSTEM or IMAGE memory pools from D-RAM to Flash ROM as SYSTEM <i>start# size</i> or IMAGE <i>start# size</i> . The <i>start#</i> is a hexadecimal number that represents the start address in D-RAM. The <i>size</i> is the size of the memory.
FRLOAD name	Loads SYSTEM or IMAGE memory pools from Flash ROM to D-RAM.
LOADALL	Loads SYSTEM and IMAGE memory pools from Flash ROM to D-RAM. You must verify that you want to perform this function.

CAUTION

All functions marked with an asterisk (*) can modify controller memory. If they are used incorrectly they will corrupt your controller.

Procedure C-10 Using FROM Utilities

- Condition**
- All personnel and unnecessary equipment are out of the workcell.
 - The BMON> prompt is displayed. Refer to Procedure C-6 .
- Step**
- 1 Press NEXT, >, until F2, INSTALL is displayed.
 - 2 Press NEXT, >, until F1, FROM is displayed.
 - 3 Press F1, FROM and press ENTER.
 - 4 **To execute a utility**, press the appropriate function key and press ENTER.

C.2.5 Memory Card Utilities

You can access the memory card (MCARD) utilities from BootROM to use a memory card. Table C-7 lists the memory card items you can use. Use Procedure C-11 to access the MCARD utilities.

Table C-7. Memory Card Items

ITEM	DESCRIPTION
EXIT	Exits to BMON>.
MCDB addr	Displays Memory Card memory address in byte/word/long word format.
MCDW addr	
MCDL addr	
MCSAVE CMOS MCSAVE DRAM MCSAVE FROM start_addr size	Saves C-MOS RAM 0.5 MB, 1.0 MB, or 2.0 MB to memory card (MC:). Saves D-RAM, always 0 – 2 MB, to memory card (MC:). Saves Flash ROM from start_addr (HEX) for the specified size (a maximum of 2 MB) to memory card (MC:)
MCLOAD CMOS* MCLOAD DRAM* MCLOAD FROM start_addr size*	Loads from memory card (MC:) to C-MOS 0.5 MB, 1.0 MB, or 2.0 MB. Loads from memory card (MC:) to D-RAM, always 0 – 2 MB. Loads from memory card (MC:) to Flash ROM from start_addr (HEX) for the specified size (a maximum of 2 MB). NOTE The area of Flash ROM on which to load must have been cleared before the MCLOAD FROM can be executed.

CAUTION

All functions marked with an asterisk (*) can modify controller memory. If they are used incorrectly they will corrupt your controller.

Procedure C-11 Using MCARD Utilities

- | | |
|------------------|---|
| Condition | <ul style="list-style-type: none"> ■ All personnel and unnecessary equipment are out of the workcell. ■ The BMON> prompt is displayed. Refer to Procedure C-6 . |
| Step | <ol style="list-style-type: none"> 1 Press NEXT, >, until F4, MCARD is displayed. 2 Press F4, MCARD, then press ENTER. 3 To execute a utility, press the appropriate function key and press ENTER. |

D EXAMPLE PROGRAMS

D

PROGRAM EXAMPLES

Topics In This Appendix	Page
Position Register Element	D-2
Position Register Value	D-2
Register Analog Input	D-3
Conditional Branching; Using Labels	D-3
Register Increment	D-4
Group Output; Wait Instruction; Pulse Instruction	D-4
Labels	D-5
Label; Jump Label; Message	D-5
Macro Instruction	D-6

This appendix contains program examples that illustrate the use of teach pendant program instructions.

Each example shows one or more program elements used in a program. **To use this section, look for the program instruction you want in the left column.** Comments follow each program.

D.1 /PROG PREG_ELE

/PROG PREG_ELE instructions and program comments are shown in Figure D-1.

Figure D-1. /PROG PREG_ELE

INSTRUCTION	DESCRIPTION
1: !POSITION REG VALUE	1. REMARK instruction, identified by an !, with the message "ARCSTART DIRECT VALUE" displayed within the program.
2: J P[1:ABOVE JOINT] 100% FINE	2. Joint move to position 1:ABOVE JOINT with 100% travel speed and FINE termination.
3: J P[2] 100% FINE	3. Joint move to position 2 with 100% travel speed and FINE termination.
4: PR[1]=LPOS	4. Position register 1 equals the current Cartesian coordinates position (x,y,z,w,p,r,config)
5: PR[1,2]=600	5. The second element of position register 1 equals 600
6: L PR[1] 100.0inch/min FINE	6. Linear move to position register 1 with 100 inches per minute travel speed and FINE termination.
7: J P[1:ABOVE JOINT] 100% FINE	7. Joint move to position 1:ABOVE JOINT with 100% travel speed and FINE termination.
/END	

D.2 /PROG PREG_VAL

/PROG PREG_VAL instructions and program comments are shown in Figure D-2.

Figure D-2. /PROG PREG_VAL

INSTRUCTION	DESCRIPTION
1: !POSITION REG VALUE	1. REMARK instruction, identified by an !, with the "POSITION REG VALUE" displayed within the program.
2: J P[1:ABOVE JOINT] 100% FINE	2. Joint move to position 1:ABOVE JOINT with 100% travel speed and FINE termination.
3: J P[2] 100% FINE	3. Joint move to position 2 with 100% travel speed and FINE termination.
4: PR[1]=LPOS	4. Position register 1 equals the current Cartesian coordinates position (X,Y,Z,w,p,r,config).
5: PR[1]=PR[1]+PR[2]	5. Position register 1 equals position register 1 plus position register 2.
6: J PR[1] 100% FINE	6. Joint move to position register 1 with 100% travel speed and FINE termination.
7: J P[1:ABOVE JOINT] 100% FINE	7. Joint move to position 1:ABOVE JOINT with 100% travel speed and FINE termination.
/END	

D.3 REGISTER ANALOG INPUT

/PROG REG_AI instructions and program comments are shown in Figure D-3.

Figure D-3. /PROG REG_AI

INSTRUCTION	DESCRIPTION
1: !REGISTER : ANALOG IN	1. REMARK instruction, identified by an !, with the message "REGISTER : ANALOG IN" displayed within the program.
2: R[1]=0	2. Register 1 equals the value 0.
3: R[2]=0	3. Register 2 equals the value 0.
4: R[1]=AI[3]	4. Register 1 equals the value of analog input 3
5: R[1]=R[1]-256	5. Register 1 equals register 1 minus 256.
6: R[2]=R[1] DIV 256	6. Register 2 equals register 1 divided by 256.
7: LBL[1:zero check]	7. The Label marks the program as the destination of a program branch. The label can have an identifier i.e. 'zero check' (LBL 1:zero check).
8: R[2]=R[2]+1	8. Register 2 equals register 2 plus 1.
9: IF R[2]=0,JMP LBL[1]	9. If instruction branches based upon the decision. If register 2 equals zero then jump to label 1 (program step 7).
10: J P[1:SAFE POSITION] 100% FINE	10. Joint move to position 1:SAFE POSITION with 100% travel speed and FINE termination.
11:J P[1:SAFE POSITION] 100% FINE	11. Joint move to position 1:SAFE POSITION with 100% travel speed and FINE termination.
/END	

D.4 CONDITIONAL BRANCHING; USING LABELS

/PROG REG_GI instructions and program comments are shown in Figure D-4.

Figure D-4. /PROG REG_GI

INSTRUCTION	DESCRIPTION
1: !REGISTER : GROUP INPUT	1. REMARK instruction, identified by an !, with the message "REGISTER : GROUP INPUT" displayed within the program.
2: LBL[1:check schd num]	2. The Label marks the program as the destination of a program branch. The label can have an identifier, i.e. 'check schd num' (LBL 1:check schd num).
3: R[1]=GI[1]	3. Register 1 equals the value of group input 1.
4: IF GI[1]>32,JMP LBL[1]	4. If instruction branches based upon the decision. If group input 1 is greater than 32 then jump to label 1 (program step 2).
5: IF GI[1]<=0,JMP LBL[1]	5. If instruction branches based upon the decision. If group input 1 is less than or equal to 0 then jump to label 1 (program step 2).
6:J P[1:ABOVE PART] 100% FINE	6. Joint move to position 1:ABOVE PART with 100% travel speed and FINE termination.
7:J P[2] 40% FINE	7. Joint move to position 2 with 40% travel speed and FINE termination.
8:J P[1:ABOVE PART] 100% FINE	8. Joint move to position 1:ABOVE PART with 100% travel speed and FINE termination.
/END	

**D.5
REGISTER
INCREMENT**

This program is an example of using the Register instructions to store the number of cycles.

R[11] is used to store the cycle count.
DI[1] is the digital input that signals a completed cycle. Refer to Figure D-5 for instructions and program comments.

Figure D-5. /PROGRAM CYCLECNT – REGISTER INCREMENT

INSTRUCTION	DESCRIPTION
1: ! This program counts cycles	1. A remark instruction, identified by an !.
2: R[11] = R[11] + DI[1]	2. Register 11 is added to the digital input. If the digital input signal is high the count is incremented. The result is stored in R[11]

**D.6
GROUP OUTPUT;
WAIT INSTRUCTION
PULSE INSTRUCTION**

This program is an example of using the Register instructions and digital input and output signal to do some external handshaking

R[11] is used to store the style type.
R[12] is used to store the error codes.
GO[11] is used to send style type to external device.
GI[11] is used to receive the error codes.
DI[11] digital input that signals style number on group output
DO[11] digital output that signals error codes on group input.

Refer to Figure D-6 for instructions and program comments.

Figure D-6. /PROGRAM SIGNAL – Group Output; WAIT and PULSE Instruction

INSTRUCTION	DESCRIPTION
1: !	1. A remark instruction, identified by an !
2: GO[11] = R[11]	2. Style number stored in register[11] is put on Group output lines 11.
3: DO[11] = PULSE ,.5sec	3. Program signals via digital output 11 that group output lines have a valid number.
4: WAIT DI[11]	4. Wait until received acknowledge is received. This signal is also used to indicate that a valid error code in on group input lines.
5: R[12] = GI[11]	5. Store error code in register 12.
6: DO[11] = PULSE ,.5sec	6. Send an acknowledgment of group input received.

D.7 LABELS

This program is an example of using the register instruction to pass values to a KAREL softpart. Refer to Figure D-7 for instructions and program comments.

Figure D-7. /PROGRAM MAIN – LABELS

INSTRUCTION	DESCRIPTION
1: ! Main program of example	1. Remark instruction identified by an !.
2: P[1] L 50mm/sec	2. Move robot to communication position.
3: CALL SIGNAL	3. Call signal program to send style number to seal equipment and get error codes back.
4: IF R[12] = 0 JMP LBL[1]	4. If no error (=0) then jump to continue.
5: JMP LBL[R[12]]	5. Jump to label of error code.
6: ABORT	6. Otherwise, abort.
7: LBL[23: BAD STYLE]	7. Label 23 bad style.
8: MESSAGE[BAD STYLE]	8. Print message on user screen that a bad style was selected.
9: JMP LBL[1]	9. Then jump over rest of error messages.
10: LBL[24: TOOL BROKEN]	10. Label 24 tool broken.
11: MESSAGE [TOOL BROKEN]	11. Print message on user screen that a broken tool was discovered.
12: LBL[1: CONTINUE]	12. Continue to other programs.

D.8 LABEL; JUMP LABEL; MESSAGE

This program is an example of using the JMP and LBL instructions. R[12] is the register that contains the error codes. It is set in program signal.

LBL[1] is the continue label

LBL[23] is the part of the program that handles bad style error codes

LBL[24] is the part of the program that handles a broken tool error code

SIGNAL is a program that communicates to external hardware and sets R[12] to the error codes received by the external hardware. Refer to Figure D-8 for instructions and program comments.

Figure D-8. /PROGRAM MAIN – LABEL; JUMP LABEL MESSAGE

INSTRUCTION	DESCRIPTION
1: REMARK Main program of example	1. Remark instruction
2: P[1] L 50mm/sec	2. Move robot to communication position.
3: CALL SIGNAL	3. Call signal program to send style number to seal equipment and get error codes back.
4: IF R[12] = 0 JMP LBL[1]	4. If no error (= 0) then jump to continue.
5: JMP LBL[R[12]]	5. Jump to label of error code.
6: ABORT	6. Else Abort.
7: LBL[23: BAD STYLE]	7. Label 23 Bad Style.
8: MESSAGE[BAD STYLE]	8. Print message on user screen that a bad style was selected.
9: JMP LBL[1]	9. Then jump over rest of error messages.
10: LBL[24: TOOL BROKEN]	10. Label 24 Tool broken.
11: MESSAGE [TOOL BROKEN]	11. Print message on user screen that a broken tool was discovered.
12: LBL[1: CONTINUE]	12. Continue to other programs.

D.9 MACRO INSTRUCTION

The following program is an example of using the interference zone macros.

ENTER ZONE 1
EXIT ZONE 1 are macro instructions that signal the other that this program is entering an interference zone. It will wait and post an error if the other machine is already there. Otherwise it will enter the zone and then leave when done. Refer to Figure D-9 for the instructions and program comments.

Figure D-9. /PROG MAIN

INSTRUCTION	DESCRIPTION
1: ! Main program of example	1. Remark instruction, identified by an !
2: P[1] L 50mm/sec	2. Move robot to position outside of zone.
3: P[2] L 50mm/sec	3. Move robot to position outside of zone.
4: ENTER ZONE 1	4. Asking to enter zone, waiting if not clear.
5: P[3] L 50mm/sec	5. Moving to points inside the zone.
6: P[4] L 50mm/sec	6. Moving to points inside the zone.
7: P[5] L 50mm/sec	7. Moving to points inside the zone.
8: EXIT ZONE 1	8. Clearing signals to zone so other equipment can use it.
9: P[6] L 50mm/sec	9. Moving to other points outside the zone.
10: P[7] L 50mm/sec	10. Moving to other points outside the zone.

E MASTERING

When you *master* a robot you define the physical location of the robot by synchronizing the mechanical information with the robot's positional information. A robot must be mastered to operate properly. Robots are usually mastered before they leave FANUC Robotics. However, it is possible that a robot might lose its mastering data and need remastering.

Topics In This Appendix	Robots	When to Use	Page
Resetting Alarms and Preparing for Mastering	All except P-200	When you power up the robot after disconnecting the pulsecoder backup batteries you might see a SRVO-062 BZAL or SRVO-038 Pulse mismatch alarm. Before mastering the robot you must reset the alarm and rotate the motor of each axis that lost battery power to prepare the robot for mastering.	E-3
Mastering to a Fixture (Fixture Position Master)	All except P-200	<ul style="list-style-type: none"> • When mastery was lost due to mechanical disassembly or repair. • When a quick master reference position was not previously set. • Method of choice for P- and A-series robots. • Used for S- and M-series robots when extreme precision is required. • Method of choice for A-series robots. • Used for S-series and M-Series robots when extreme precision is required. 	E-6
Zero Degree Mastering	M-series S-series P-series	<ul style="list-style-type: none"> • When mastery was lost due to mechanical disassembly or repair. • When a quick master reference position was not previously set. • Method of choice for S- and M-series robots when extreme precision is not required. 	E-8
Single Axis Mastering	S-series M-series P-series	When mastery was lost due to mechanical disassembly or repair of a single axis (usually due to motor replacement).	E-10
Quick Mastering	All	To retrieve mastering data that has been stored as a quick master reference position when mastery is lost due to an electrical or software problem. Do not use if mastery was lost due to mechanical disassembly or repair.	E-13

Before you master the robot, you must clear any faults that prevent servo power from being restored or that prevent mastering completion.

Use Procedure E-1 to clear common faults related to mastering and to prepare the robot for mastering. For more detailed information on fault recovery, refer to the *FANUC Robotics SYSTEM R-J2 Controller Series Electrical Connection and Maintenance Manual*.

If you are using a FANUC Robotics A-series or P-series robot you should use a mastering fixture to master your robot. Refer to the *FANUC Robotics SYSTEM R-J2 Controller Mechanical Connection and Maintenance Manual* specific to your robot model for procedures on how to set up and use a mastering fixture.

NOTE If you are using a FANUC Robotics P-series robot, and you have witness marks scored onto your robot, then you can master the robot to zero degrees. Refer to the *Mechanical Service Manual* specific to your robot model for more information

If you are using a FANUC Robotics M-series or S-series robot you can either master to a fixture or you can master to zero degrees. Refer to the *FANUC Robotics SYSTEM R-J2 Controller Mechanical Connection and Maintenance Manual* specific to your robot model for more information on mastery using a fixture.

Quick mastering is a convenient way to master an M-series, P-series or S-series robot **after** you have recorded a reference position. You cannot quick master a robot unless the reference position was taught before mastering was lost.

**CAUTION**

Record the quick master reference position after the robot is installed to preserve the factory mastering settings for future remastering.

E.1 RESETTING ALARMS AND PREPARING FOR MASTERING

When you power up the robot after disconnecting the pulsecoder backup batteries you might see a SRVO-062 BZAL or SRV0-038 Pulse mismatch alarm. Before mastering the robot you must reset the alarm and rotate the motor of each axis that lost battery power to prepare the robot for mastering.

Use Procedure E-1 to reset these alarms and prepare the robot for mastering.

Procedure E-1 Preparing the Robot for Mastering

- Condition** ■ You see a SRVO-062 BZAL or SRVO-038 Servo mismatch alarm.
- Step**
- 1 Replace the robot batteries with four new 1.5 volt alkaline batteries, size D. Observe the direction arrows in the battery box for proper orientation of the batteries.
 - 2 Press MENUS.
 - 3 Select SYSTEM.
 - 4 Press F1, [TYPE].
 - 5 Select Master/Cal.

If Master/Cal is not listed on the [TYPE] menu, do the following; otherwise, continue to Step 6.

 - a Select VARIABLE from the [TYPE] menu.
 - b Move the cursor to \$MASTER_ENB.
 - c Press the numeric key “1” and then press ENTER on the teach pendant.
 - d Press F1, [TYPE].
 - e Select Master/Cal. You will see a screen similar to the following.

```

SYSTEM Master/Cal                                JOINT 10%
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE

Press 'ENTER' or number key to select.

[ TYPE ]   LOAD   RES_PCA                       DONE
  
```

⚠ WARNING
 For M-6i (ARC Mate 100i), M-16i (ARC Mate 120i), M-16iL (ARC Mate 120iL), S-6 (ARC Mate 100), and S-12 (ARC Mate 120) robots, setting TORQUE to OFF using the TORQUE function key on the SYSTEM Master/Cal screen releases the robot brakes. When the brakes are released, the robot arm will drop suddenly unless it is supported. **DO NOT** use this function key unless instructed to do so, otherwise, personnel could be injured and equipment damaged.

6 Press F3, RES_PCA. You will see a screen similar to the following.

```

SYSTEM Master/Cal                               JOINT 10%
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.

Reset pulse coder alarm? [NO]

[ TYPE ]                                     YES      NO
  
```

7 Press F4, YES. You will see a screen similar to the following.

```

SYSTEM Master/Cal                               JOINT 10%
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE
Pulse coder alarm reset!

[ TYPE ]   LOAD      RES_PCA                DONE
  
```

**WARNING**

For M-6i (ARC Mate 100i), M-16i (ARC Mate 120i), M-16iL (ARC Mate 120iL), S-6 (ARC Mate 100), and S-12 (ARC Mate 120) robots, setting TORQUE to OFF using the TORQUE function key on the SYSTEM Master/Cal screen releases the robot brakes. When the brakes are released, the robot arm will drop suddenly unless it is supported. **DO NOT** use this function key unless instructed to do so, otherwise, personnel could be injured and equipment damaged.

- 8 If the SRVO-062 alarm is still present; there is a battery, cable or pulsecoder problem. Refer to the *FANUC Robotics SYSTEM R-J2 Controller Series Electrical Connection and Maintenance Manual* for further information.
- 9 If a SRVO-038 alarm is present at this time, repeat Step 6 to reset it. It is not necessary to cold start the robot after resetting to clear this alarm.
- 10 Rotate each axis that lost battery power by at least one motor revolution in either direction.
 - a Jog each **rotary axis** at least twenty degrees.
 - b Jog each **linear axis** at least thirty millimeters.
- 11 Perform any of the mastering procedures from the MASTER/CAL menu.

E.2 MASTERING TO A FIXTURE (FIXTURE POSITION MASTER)

When you master to a fixture, you use a mastering fixture to align the robot axes and then record the position. You can master any robot to a fixture. If you have a P-series or A-series robot, you **must** master it to a fixture.

NOTE If you have an M-6i (ARC Mate 100i), S-6 (ARC Mate 100), or S-12 (ARC Mate 120) robot, **do not use** Procedure E-2 . These robots require that servo power be removed and brakes released in order to use the fixture. Refer to the *FANUC Robotics SYSTEM R-J2 Controller Mechanical Connection and Maintenance Manual* for detailed procedures on mastering these robots to a fixture.

Use Procedure E-2 to master to a fixture.

Procedure E-2 Mastering to a Fixture (for all robots except S-6 and S-12)

- | | |
|------------------|---|
| Condition | <ul style="list-style-type: none"> ■ You have the appropriate mastering fixture for your robot. ■ You have cleared any servo faults that prevent you from jogging the robot. ■ You have jogged each axis that has lost mastery at least one motor turn. (Procedure E-1) |
| Step | <ol style="list-style-type: none"> 1 Install the mastering fixture on the robot and jog the robot into mastering position. Refer to the <i>FANUC Robotics SYSTEM R-J2 Controller Mechanical Connection and Maintenance Manual</i> specific to your robot model for the procedures on how to set up and use a mastering fixture. 2 Press MENUS. 3 Select SYSTEM. 4 Press F1, [TYPE]. 5 Select Master/Cal. <p>If Master/Cal is not listed on the [TYPE] menu, do the following; otherwise, continue to Step 6.</p> <ol style="list-style-type: none"> a Select VARIABLE from the [TYPE] menu. b Move the cursor to \$MASTER_ENB. c Press the numeric key “1” and then press ENTER on the teach pendant. d Press F1, [TYPE]. |

- e Select Master/Cal. You will see a screen similar to the following.

```

SYSTEM Master/Cal                               JOINT 10%

1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE

Press 'ENTER' or number key to select.

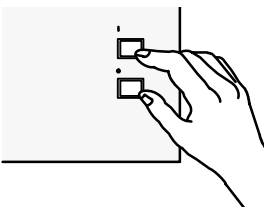
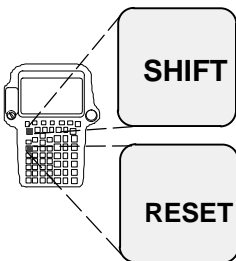
[ TYPE ]   LOAD   RES_PCA                       DONE

```

**WARNING**

For M-6i (ARC Mate 100i), M-16i (ARC Mate 120i), M-16iL (ARC Mate 120iL), S-6 (ARC Mate 100), and S-12 (ARC Mate 120) robots, setting TORQUE to OFF using the TORQUE function key on the SYSTEM Master/Cal screen releases the robot brakes. When the brakes are released, the robot arm will drop suddenly unless it is supported. DO NOT use this function key unless instructed to do so, otherwise, personnel could be injured and equipment damaged.

- 6 Select Fixture Position Master.
- 7 Select Calibrate.
- 8 Press F4, YES.
- 9 Perform a cold start.
 - a Turn off the robot.
 - b Press and continue pressing the SHIFT and RESET key on the teach pendant.
 - c While still pressing SHIFT and RESET, press the ON button on the operator panel.



E.3 ZERO DEGREE MASTERING

When you master to zero degrees, you position all axes at their zero degree witness marks and record the zero degree position. You can master any M-series or S-series robot to zero degrees.

NOTE If you are mastering an S-420iR robot, you cannot use this mastering method. Use one of the other mastering methods described in Sections E.2, E.4, and E.5.

Use Procedure E-3 to master to zero degrees.

Procedure E-3 Mastering to Zero Degrees (for all robots except S-420iR)

- Condition**
- You have cleared any servo faults that prevent you from jogging the robot.
 - You have jogged each axis that has lost mastery at least one motor turn. (Procedure E-1)

- Step**
- 1 Using the joint coordinate system, jog each axis of the robot to the zero degree witness mark. Refer to the *FANUC Robotics SYSTEM R-J2 Controller Mechanical Connection and Maintenance Manual* specific to your robot model for the location of the witness marks.
 - 2 Press **MENUS**.
 - 3 Select **SYSTEM**.
 - 4 Press **F1**, [**TYPE**].
 - 5 Select **Master/Cal**.

If Master/Cal is not listed on the [TYPE] menu, do the following; otherwise, continue to Step 6.

- a** Select **VARIABLE** from the [**TYPE**] menu.
- b** Move the cursor to **\$MASTER_ENB**.
- c** Press the numeric key “1” and then press **ENTER** on the teach pendant.
- d** Press **F1**, [**TYPE**].

- e Select Master/Cal. You will see a screen similar to the following.

```

SYSTEM Master/Cal                               JOINT 10%
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE

Press 'ENTER' or number key to select.

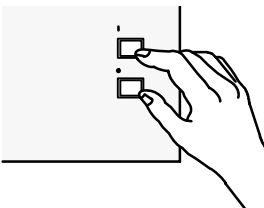
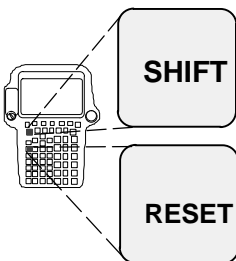
[ TYPE ]   LOAD   RES_PCA                       DONE

```

**WARNING**

For M-6i (ARC Mate 100i), M-16i (ARC Mate 120i), M-16iL (ARC Mate 120iL), S-6 (ARC Mate 100), and S-12 (ARC Mate 120) robots, setting TORQUE to OFF using the TORQUE function key on the SYSTEM Master/Cal screen releases the robot brakes. When the brakes are released, the robot arm will drop suddenly unless it is supported. **DO NOT** use this function key unless instructed to do so, otherwise, personnel could be injured and equipment damaged.

- 6 Select Zero Position Master.
- 7 Press F4, YES. Mastering will be performed automatically.
- 8 Select Calibrate.
- 9 Press F4, YES.
- 10 Perform a cold start.
 - a Turn off the robot.
 - b Press and continue pressing the SHIFT and RESET key on the teach pendant.
 - c While still pressing SHIFT and RESET, press the ON button on the operator panel.



E.4 SINGLE AXIS MASTERING

You can master a single axis of an M-series or S-series robot when mastery was lost due to mechanical disassembly or repair of a single axis, usually due to motor replacement.

Use Procedure E-4 to master a single axis.

Procedure E-4 Mastering a Single Axis

NOTE For the S-420iR robot, when the witness marks are aligned for axes J2 and J3, the joint angles are as follows:

- J2: -15°
- J3: -15°

- | | |
|------------------|--|
| Condition | <ul style="list-style-type: none"> ■ You have cleared any servo faults that prevent you from jogging the robot. ■ You have jogged each axis that has lost mastery at least one motor turn. (Procedure E-1) |
| Step | <ol style="list-style-type: none"> 1 Using the joint coordinate system, jog the unmastered axis of the robot to the zero degree witness mark. Refer to the <i>FANUC Robotics SYSTEM R-J2 Controller Mechanical Connection and Maintenance Manual</i> specific to your robot model for the location of the witness marks. 2 Press MENUS. 3 Select SYSTEM. 4 Press F1, [TYPE]. 5 Select Master/Cal. <p style="margin-left: 20px;">If Master/Cal is not listed on the [TYPE] menu, do the following; otherwise, continue to Step 6.</p> <ol style="list-style-type: none"> a Select VARIABLES from the [TYPE] menu. b Move the cursor to \$MASTER_ENB. c Press the numeric key “1” and then press ENTER on the teach pendant. d Press F1, [TYPE]. |

- e Select Master/Cal. You will see a screen similar to the following.

```

SYSTEM Master/Cal                               JOINT 10%

1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE

Press 'ENTER' or number key to select.

[ TYPE ]   LOAD   RES_PCA                       DONE

```

**WARNING**

For M-6i (ARC Mate 100i), M-16i (ARC Mate 120i), M-16iL (ARC Mate 120iL), S-6 (ARC Mate 100), and S-12 (ARC Mate 120) robots, setting TORQUE to OFF using the TORQUE function key on the SYSTEM Master/Cal screen releases the robot brakes. When the brakes are released, the robot arm will drop suddenly unless it is supported. DO NOT use this function key unless instructed to do so, otherwise, personnel could be injured and equipment damaged.

- 6 Select 4, Single Axis Master. You will see a screen similar to the following.

```

SINGLE AXIS MASTER                               JOINT 10%
                                                1/9

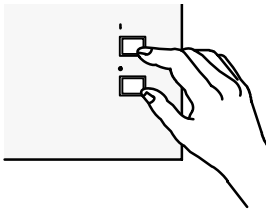
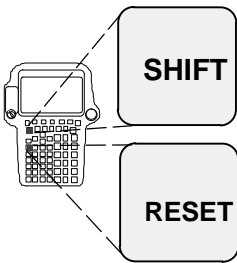
ACTUAL POS      (MSTR POS)      (SEL) [ST]
J1    0.000      ( 0.000)      (0) [2]
J2    3.514      ( 35.000)     (0) [0]
J3   -7.164     (-100.000)    (0) [2]
J4 -357.366     ( 0.000)      (0) [2]
J5   -1.275     (-80.000)    (0) [2]
J6    4.571     ( 0.000)      (0) [2]
E1    0.000     ( 0.000)      (0) [0]
E2    0.000     ( 0.000)      (0) [0]
E3    0.000     ( 0.000)      (0) [0]

[ TYPE ]                       GROUP   EXEC

```

- 7 Move the cursor to the MSTR POS column for the unmastered axis and press the "0" key.
- 8 Continuously press and hold the DEADMAN switch and turn the teach pendant ON/OFF switch to ON.

- 9** Move the cursor to the SEL column for the unmastered axis and press the numeric key “1.”
- 10** Press ENTER.
- 11** Press F5, EXEC. Mastering will be performed automatically.
- 12** Press PREV.
- 13** Select Calibrate.
- 14** Press F4, YES.
- 15** Perform a cold start.
 - a** Turn off the robot.
 - b** Press and continue pressing the SHIFT and RESET key on the teach pendant.
 - c** While still pressing SHIFT and RESET, press the ON button on the operator panel.



E.5 QUICK MASTERING

Quick mastering allows you to minimize the time required to remaster the robot using a reference position you established when the robot was properly mastered. You cannot quick master the robot unless you have previously recorded this quick master reference position.

Record the quick master reference position when the robot is properly mastered. The best time to record the quick master reference position is when the robot is still factory-mastered.

If you lose mastery due to an electrical or software problem, you can use this reference position to master the robot in a minimum amount of time. If you lose mastery due to mechanical disassembly or repair, you must master to a fixture or perform zero degree mastering.

You can define a quick master reference position and perform quick mastering on any robot model.

Use Procedure E-5 to record the quick master reference position. Use Procedure E-6 to quick master the robot.

CAUTION

Record the quick master reference position after the robot is installed to preserve the factory mastering settings for future remastering.

Procedure E-5 Recording the Quick Master Reference Position

NOTE For the S-420iR robot, when the witness marks are aligned for axes J2 and J3, the joint angles are as follows:

- J2: -15°
- J3: -15°

- | | |
|------------------|---|
| Condition | ■ The robot is properly mastered. |
| Step | <ol style="list-style-type: none"> 1 Align the witness marks on the robot. This is the zero position, which will be the quick master reference position. Refer to the <i>FANUC Robotics SYSTEM R-J2 Controller Mechanical Connection and Maintenance Manual</i> specific to your robot model for the location of the witness marks. 2 Press MENUS. 3 Select SYSTEM. 4 Press F1, [TYPE]. |

5 Select Master/Cal.

If Master/Cal is not listed on the [TYPE] menu, do the following; otherwise, continue to Step 6.

- a** Select VARIABLE from the [TYPE] menu.
- b** Move the cursor to \$MASTER_ENB.
- c** Press the numeric key “1” and then press ENTER on the teach pendant.
- d** Press F1, [TYPE].
- e** Select Master/Cal. You will see a screen similar to the following.

```

SYSTEM Master/Cal                               JOINT 10%

1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE

Press 'ENTER' or number key to select.

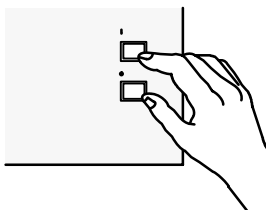
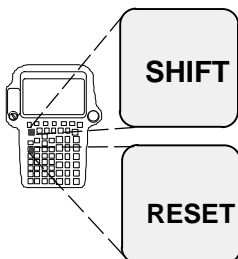
[ TYPE ]   LOAD   RES_PCA                       DONE

```

**WARNING**

For M-6i (ARC Mate 100i), M-16i (ARC Mate 120i), M-16iL (ARC Mate 120iL), S-6 (ARC Mate 100), and S-12 (ARC Mate 120) robots, setting TORQUE to OFF using the TORQUE function key on the SYSTEM Master/Cal screen releases the robot brakes. When the brakes are released, the robot arm will drop suddenly unless it is supported. **DO NOT** use this function key unless instructed to do so, otherwise, personnel could be injured and equipment damaged.

Set quick master ref? [NO]



- 6** Move the cursor to SET QUICK MASTER REF and press ENTER.
- 7** Press F4, YES.
- 8** Perform a cold start.
 - a** Turn off the robot.
 - b** Press and continue pressing the SHIFT and RESET key on the teach pendant.
 - c** While still pressing SHIFT and RESET, press the ON button on the operator panel.

Procedure E-6 Quick Mastering the Robot

- Condition**
- The robot has lost mastery due to an electrical or software problem.
- NOTE** If the robot has lost mastery due to mechanical disassembly or repair, you cannot perform this procedure. In this case, master to a fixture or master to zero degrees to restore robot mastering.
- The quick master reference position was recorded before the robot lost mastery.
 - You have cleared any servo faults that prevent you from jogging the robot.
 - You have jogged each axis that has lost mastery at least one motor turn. See Procedure E-1 .
- Step**
- 1 Jog the robot to the quick master reference position (zero degree position).
 - 2 Press **MENUS**.
 - 3 Select **SYSTEM**.
 - 4 Press **F1**, **[TYPE]**.
 - 5 Select **Master/Cal**.
- If Master/Cal is not listed on the [TYPE] menu, do the following; otherwise, continue to Step 6.**
- a** Select **VARIABLE** from the **[TYPE]** menu.
 - b** Move the cursor to **\$MASTER_ENB**.
 - c** Press the numeric key “1” and then press **ENTER** on the teach pendant.
 - d** Press **F1**, **[TYPE]**.

- e Select Master/Cal. You will see a screen similar to the following.

⚠ WARNING
 For M-6i (ARC Mate 100i), M-16i (ARC Mate 120i), M-16iL (ARC Mate 120iL), S-6 (ARC Mate 100), and S-12 (ARC Mate 120) robots, setting TORQUE to OFF using the TORQUE function key on the SYSTEM Master/Cal screen releases the robot brakes. When the brakes are released, the robot arm will drop suddenly unless it is supported. **DO NOT** use this function key unless instructed to do so, otherwise, personnel could be injured and equipment damaged.

```

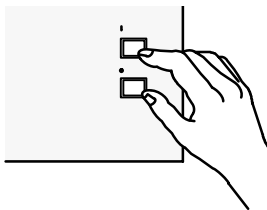
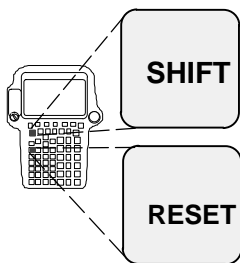
SYSTEM Master/Cal                               JOINT 10%
-----
1 FIXTURE POSITION MASTER
2 ZERO POSITION MASTER
3 QUICK MASTER
4 SINGLE AXIS MASTER
5 SET QUICK MASTER REF
6 CALIBRATE

Press 'ENTER' or number key to select.

[ TYPE ]   LOAD   RES_PCA                       DONE
  
```

- 6 Move the cursor to QUICK MASTER and press ENTER.
- 7 Press F4, YES.
- 8 Move the cursor to CALIBRATE and press ENTER
- 9 Press F4, YES.
- 10 Perform a cold start.
 - a Turn off the robot.
 - b Press and continue pressing the SHIFT and RESET key on the teach pendant.
 - c While still pressing SHIFT and RESET, press the ON button on the operator panel.

Quick master? [NO]

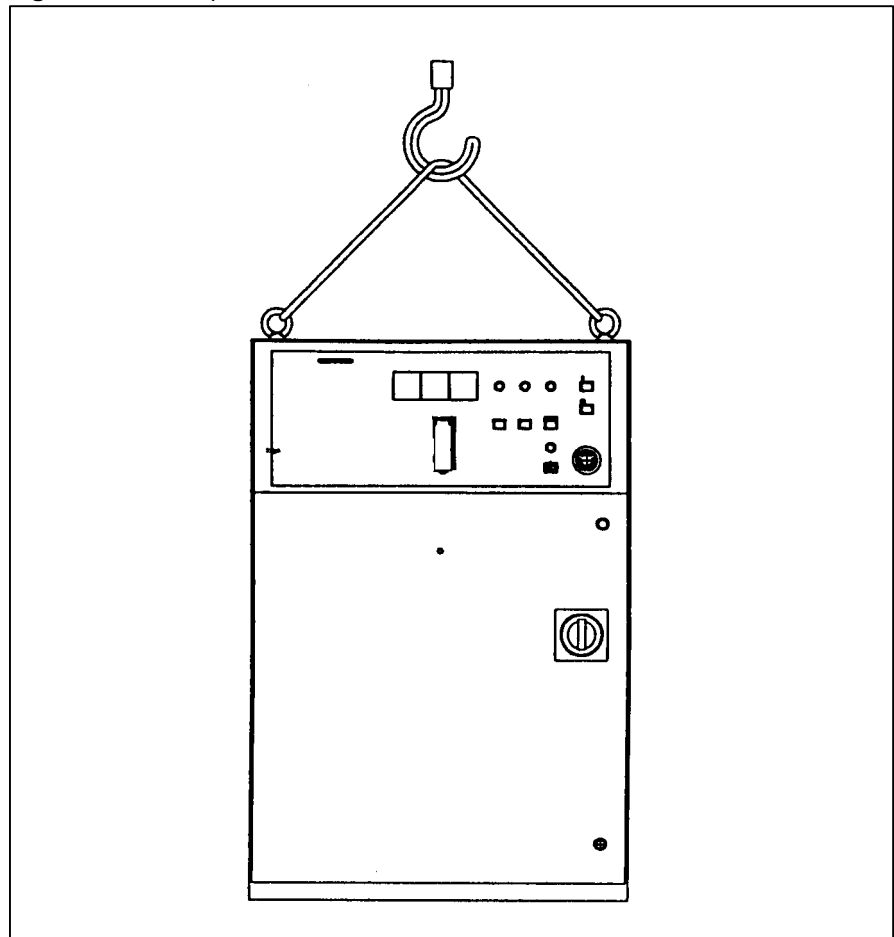


Topics In This Appendix		Page
Transportation	This section includes information on transporting and installing an R-J2 controller.	F-1

F.1 TRANSPORTATION

The controller is transported by a crane. Attach a rope to eye bolts at the top of the controller, as shown in Figure F-1.

Figure F-1. Transportation



Use Procedure F-1 to eliminate a UOP configuration.

Use Procedure F-2 to reconfigure digital I/O signals.

Use Procedure F-3 to reset an overtravel.

Procedure F-1 Eliminating a UOP Configuration

To move the robot without hooking up the UOP Interface cable or to eliminate UOP configuration complete the following

- | | |
|-------------|---|
| Step | <ol style="list-style-type: none">1 Press MENUS.2 Select I/O.3 Press F1, [TYPE].4 Select UOP.5 Press F2, Config.6 Set all UO and UI configurations to:<ol style="list-style-type: none">a Rack: 0b Slot: 0c Point: 0 |
|-------------|---|

NOTE Refer to Section 3.3 for more detail on configuring UOPs.

Procedure F-2 Reconfiguring Digital I/O Signals

- | | |
|-------------|---|
| Step | <ol style="list-style-type: none">1 Press MENUS.2 Select I/O.3 Press F1, [TYPE].4 Select Digital.5 Reconfigure DIs and DOs to include previously assigned UOP I/O points. |
|-------------|---|

Procedure F-3 Resetting Overtravel

- | | |
|-------------|--|
| Step | <ol style="list-style-type: none">1 Press and hold the SHIFT key.2 Press the RESET switch on the operator panel or the RESET key on the teach pendant.3 Hold SHIFT key while jogging the robot out of the overtravel position. (Manually move an axis that has overtraveled into the operating range while pressing the shift key on the teach pendant.) |
|-------------|--|

GLOSSARY

A

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

absolute pulse code system. A positional information system for servomotors that relies on battery-backed RAM to store encoder pulse counts when the controller is powered down. This system powers up calibrated.

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

AI. Analog input.

AO. Analog output.

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

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

alphanumeric. Data that are both alphabetical and numeric.

AMPS, Amperage amount.

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

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

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

APC. See absolute pulse code system.

APC motor. See servomotor.

application program. The set of instructions that defines the specific intended tasks of robots and robot systems to make them reprogrammable and multifunctional. These programs can be initiated and changed by the robot user.

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

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

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

automatic mode. The robot state in which automatic operation can be initiated.

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

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

B

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

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

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

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

binary. A numbering system that uses only 0 and 1.

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

bps. Bits per second.

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

bus. A channel along which data can be sent.

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

bus scan. One complete communications cycle on the serial bus.

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

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

C

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

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

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

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

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

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

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

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

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

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

CMOS RAM. Complementary metal-oxide semiconductor read-access memory. A read/write memory that consumes little power but requires a battery in order to retain content upon a loss of power.

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

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

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

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

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

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

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

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

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

control, open-loop. An operation where the computer applies control directly to the process without manual intervention.

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

coordinate system. See Cartesian coordinate system.

CPU. See central processing unit.

CRT. See cathode ray tube.

CRT/KB. Cathode ray tube/keyboard. An optional interface device for the SYSTEM R-J robot system. The CRT/KB is used for some robot operations and for entering programs. It can be built into the controller cabinet or it can be a remote device that attaches to the controller via a cable.

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

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

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

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

D

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

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

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

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

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

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

DC. Abbreviation for direct current.

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

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

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

default. The value, display, function or program automatically selected if the user has not specified a choice.

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

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

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

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

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

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

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

directory. A listing of the files stored on a device.

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

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

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

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

E

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

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

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

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

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

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

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

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

EOAT. See end of arm tooling, tool.

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

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

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

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

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

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

F

faceplate. The tool mounting plate of the robot.

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

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

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

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

filter. A device to suppress interference that would appear as noise.

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

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

format. To set up or prepare a floppy disk so it can be used to store data in a specific system.

G

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

Genius I/O bus. The serial bus that provides communications between blocks, controllers, and other devices in the system esp. W.R.I. GE FANUC Genius I/O.

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

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

gun. See applicator.

H

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

hard-wire. To connect electric components with solid metallic wires.

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

hazardous motion. Unintended or unexpected robot motion that can cause injury.

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

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

I

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

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

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

index register. A memory device containing an index.

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

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

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

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

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

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

input processing time. The time required for input data to reach the microprocessor.

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

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

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

interactive. Refers to applications where a user communicates with a computer program via a terminal by entering data and receiving responses from the computer.

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

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

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

I/O. Abbreviation for input/output or input/output control.

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

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

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

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

ISO. The International Standards Organization that establishes the ISO interface standards.

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

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

J

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

JOGFRAME. A jog coordinate system the user defines to make the robot jog the best way possible for a specific application may be different from world coordinate frame.

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

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

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

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

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

K

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

KAREL. The programming language developed for robots by the FANUC Robotics Corporation.

L

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

LCD. See liquid crystal display.

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

LED. See Light Emitting Diode.

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

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

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

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

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

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

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

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

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

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

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

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

M

mA. See milliamper.

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

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

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

magnetic disk. A metal or plastic floppy disk that looks like a phonograph record whose surface can store data in the form of magnetized spots.

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

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

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

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

memory. A device or media used to store information in a form that can be retrieved and is understood by the computer or controller hardware. Memory on the R-J system includes CMOS RAM, Flash ROM and RAM disk (optional).

menu. A list of options displayed on the teach pendant screen.

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

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

microsecond. One millionth (0.000001) of a second

milliamper. One one-thousandth of an ampere. Abbreviated mA.

millisecond. One thousandth of a second. Abbreviated msec.

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

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

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

N

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

non-volatile memory. Memory capable of retaining its stored information when power is turned off.

O

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

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

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

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

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

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

operating system. Lowest level system monitor program.

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

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

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

optional features. Additional capabilities available at a cost above the base price.

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

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

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

OT. See overtravel.

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

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

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

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

override. See general override.

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

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

P

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

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

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

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

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

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

PCB. See printed circuit board.

pendant. See teach pendant.

PLC. See programmable logic controller or cell controller.

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

production mode. See automatic mode.

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

programmable controller. See programmable logic controller or cell controller.

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

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

Q

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

R

RAM. See Random Access Memory.

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

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

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

RI. Robot input.

RO. Robot output.

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

Read Only Memory. A digital memory containing a fixed pattern of bits that generally cannot be altered by the user.

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

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

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

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

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

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

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

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

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

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

ROM. See Read Only Memory.

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

S

saving data. Storing program data in CMOS RAM or the RAM disk (optional).

SI. System input.

SO. System output.

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

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

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

servomotor. An electric motor that is controlled to produce precision motion. Also called a “smart” motor.

signal. The event, phenomenon, or electrical quantity that conveys information from one point to another.

significant bit. A bit that contributes to the precision of a number. These are counted starting with the bit that contributes the most value, of “most significant bit,” and ending with the bit that contributes the least value, or “least significant bit.”

Standard Operator Panel (SOP). A panel located on the front of the R-J controller that is made up of buttons, keyswitches, and connector ports.

state. The on or off condition of current to and from and input or output device.

statement. See instruction.

storage device. Any device that can accept, retain, and read back one or more times. The available storage devices are CMOS RAM, RAM disk (optional) and floppy disks.

system variable. An element that stores data used by the R-J system to indicate such things as robot specifications, application requirements, and the current status of the system.

T

TCP. See tool center point.

teaching. Generating and storing a series of positional data points effected by moving the robot arm through a path of intended motions.

teach mode. 1. The mode of operation in which a robot is instructed in its motions, usually by guiding it through these motions using a teach pendant. 2. The generation and storage of positional data. Positional data can be taught using the teach pendant to move the robot through a series of positions and recording those positions for use by an application program.

teach pendant. 1. A hand-held device used to instruct a robot, specifying the character and types of motions it is to undertake. Also known as teach box, teach gun. 2. A portable device, consisting of an LCD display and a keypad, that serves as a user interface to the KAREL system and attaches to the controller via a cable. The teach pendant is used for robot operations such as jogging the robot, teaching and recording positions, and testing and debugging programs.

termination type. Feature that controls the blending of robot motion between segments.

tool. A term used loosely to define something mounted on the end of the robot arm, for example, a hand, gripper, or an arc welding torch.

tool center point. 1. The location on the end-effector or tool of a robot hand whose position and orientation define the coordinates of the controlled object. 2. Reference point for position control, that is, the point on the tool that is used to teach positions. Abbreviated TCP.

Tool Frame. The Cartesian coordinate system that has the position of the TCP as its origin to set. The z-axis of the tool frame indicates the approach vector for the tool.

transducer. A device for converting energy from one form to another.

U

UOP. See user operator panel.

user frame. The Cartesian coordinate system that the user can define for a specific application. The default value of the User Frame is the World Frame. All positional data is recorded relative to User Frame.

User Operator Panel. User-supplied control device used in place of or in parallel with the operator panel supplied with the controller. Abbreviated UOP.

V

variable. A quantity that can assume any of a given set of values.

variance. The difference between the expected (or planned) and the actual, also statistics definitions.

vision system. A device that collects data and forms an image that can be interpreted by a robot computer to determine the position or to “see” an object.

volatile memory. Memory that will lose the information stored in it if power is removed from the memory circuit device.

W

warning device. An audible or visible device used to alert personnel to potential safety hazards.

work envelope. The volume of space that encloses the maximum designed reach of the robot manipulator

including the end effector, the workpiece, and the robot itself. The work envelope can be reduced or restricted by limiting devices. The maximum distance the robot can travel after the limit device is actuated is considered the basis for defining the restricted work envelope.

write. To deliver data to a medium such as storage.

A

- A-520i, 1-4
- abort, instructions, 6-127
- \$accel, \$TCPPSPEED structure, 10-180
- \$accel, \$TCPPSPEED, 10-180
- acceleration override, motion option, 6-35
- accessing BootROM, C-14
- adding, instructions in a program, 5-11
- adjust utility, in shape generation, 10-71
- advanced functions
 - Collision Guard, 10-141
 - program adjust schedules, 10-3
 - space check function, 10-26
- alarm
 - code monitoring, error recovery, 10-154
 - digital input, error recovery, 10-156
 - user, setup, 3-155, 3-156, 3-157
- alarm code monitoring
 - error recovery, 10-154
 - error recovery feature, 10-149
 - maximum number of alarms, 10-154
 - setup, 10-154
- alarm log, displaying the, A-3
- alarm recovery screen, error recovery feature, 10-149
- alarms, user, error recovery, 10-156
- Allen Bradley Remote, network, 3-4, 3-49
- analog
 - I/O, 3-3, 3-4
 - channel, 3-5
 - comments, 3-6
 - rack, 3-5
 - slot, 3-5
 - modular I/O board layout, 3-6
 - process I/O board layout, 3-5
- analog I/O, instructions, 6-103
- application setup, screen item, C-4
- application teach pendant program files, backing up, 9-53
- applying, brakes, 3-151
- approval DI, error recovery, 10-149
- approval DI is on, manual function detail screen, error recovery, 10-162
- APSH Error Codes, A-17
- arc start instruction, 10-158
- arc welding, system setup monitors, 10-161, 10-162
- ATPERCH, UOP output signals, 3-39
- attitude, positional data conversion, coordinates offset function, 10-111
- [ATTR], 6-5, 6-6, 6-7, 6-8
- attributes, program, 6-5, 6-6, 6-7, 6-8
- AUTO, MODE SELECT switch, 7-26
- AUTO mode
 - errors, 1-25
 - locking, 1-25
 - MODE SELECT switch, 1-24, 7-18, 7-22, 7-23, 7-24, 7-25, 7-27, 7-28
 - European controllers, 1-22
 - program activation, 1-24
 - robot speed, 1-24
 - safety equipment, 1-24
 - system variables, 1-25
- Auto Normal, utility, 10-169
- auto start max count, setup item, error recovery, 10-152
- auto start max count R[], setup item, error recovery, 10-152
- AUTO STOP, Control Reliable option, 8-15
- Auto TCP
 - Auto Setup Menu, 12-1, 12-13
 - Automatic position generation, 12-1, 12-14
 - Automatically generating positions, 12-1, 12-16
 - Manually defining positions, 12-1, 12-14
 - Tool Orientation, 12-1, 12-13
 - hardware installation, 12-1, 12-3
 - sensor I/O signal cables, 12-5
 - sensor setup, 12-1, 12-7
 - string sensor, 12-3
 - TCP attachment device, 12-3
 - testing data ports, 12-1, 12-10
 - hardware requirements, 12-1, 12-3
 - overview, 12-1, 12-2
 - TCP orientation setup, 12-1, 12-11
- Auto TCP Points, 12-14
- automatic error recovery, normal operation without resume program execution, timing diagram, 10-165
- automatic error recovery enabled, manual function detail screen, error recovery, 10-162
- Automatic Mode. *See* AUTO mode

automatic start, error recovery feature, 10–149

automatic start feature, setup item, error recovery, 10–152

Automatic Tool Center Point. *See* Auto TCP

axes

- extended, 1–31
- number of, 1–31
- robot, 1–31

Axis

- Definition of, 1–3
- Major Axes, 1–3

axis control board, RI/RO signals, 3–49

axis limits

- hardstops, 3–145
- limit switches, 3–145
- lower limits, 3–146
- saving limits, 3–146
- setup, 3–145
- software settings, 3–145
- upper limits, 3–146

axis location, positional data conversion, coordinates

- offset function, 10–111

B

background

- editing, 5–15, 5–30
- system variable, 5–15, 5–30

background editing, 5–30

backing up

- a controller, 9–66
- application files, 9–56
- files from FILE menu, 9–55
- program files, 9–55
- system files, 9–55

backing up program, system, and application files, 9–53

backplane

- 3–slot, 1–32
- 5–slot, 1–32
- controller, 1–32

backward, testing, 7–12

basic digital I/O module, DIP switches, distributed I/O, 3–47

basic digital I/O unit

- distributed I/O, setting up, 3–49

- Model B modular I/O, configuring DIP switches, 3–47

BATALM, UOP output signals, 3–39

battery–backed, memory. *See* CMOS RAM

blend–in angle setup, shape generation

- circle schedule data, 10–56
- hexagon schedule data, 10–58
- rectangle schedule data, 10–61
- slot schedule data, 10–64

blend–in distance setup, shape generation

- circle schedule data, 10–56
- hexagon schedule data, 10–58
- rectangle schedule data, 10–61
- slot schedule data, 10–64

BMON, C–12

boot monitor. *See* BMON

BootROM, C–1

- accessing, C–14

bootrom, using utilities, C–14

brake on hold, setup, 3–151

brake timers, setup, 3–148

brakes

- applying, 3–151
- enabling. *See* applying

branching, instructions, 1–33, 6–3, 6–105

built-in, CRT/KB, B–1

BUSY, UOP output signals, 3–39

C

cables, sensor I/O signal, Auto TCP, 12–5

call, instruction. *See* subprogram call

Cartesian, coordinate system, 3–78

cathode ray tube/keyboard. *See* CRT/KB

CD Error Codes, A–19

CE MARK. *See* European controllers

cell controller, PLC I/O, 3–63

center position register, shape generation setup, 10–51

CHANGE RTCP FRAME, FCTN menu item, 2–13

channel, I/O, analog, 3–5

checking, file memory, 9–65

circle schedule data, shape generation

- blend–in angle setup, 10–56

- blend-in distance setup, 10-56
- circle type setup, 10-55
- comment, 10-55
- diameter setup, 10-55
- overlap angle setup, 10-56
- schedule number setup, 10-55
- speed setup, 10-55
- start axis setup, 10-56
- x-work angle setup, 10-56
- y-work angle setup, 10-56
- circle type setup, shape generation, circle schedule data, 10-55
- circles, shape generation, 10-50, 10-53
- circular
 - motion, 1-31
 - motion type, 6-14, 6-15
 - recording, 6-14
 - via position, 6-15
 - orientation control at intermediate via point, 6-15
- circular motion, speed, 6-24
- clear, screen item, C-13
- CLEAR_RESUME_PROG, instruction, 10-158
- clearing a user frame, 3-114
- CMCC Error Codes, A-21
- CMDENBL, UOP output signals, 3-39
- CMND Error Codes, A-22
- CMOS RAM, 1-32, 9-2
 - amount of, 1-32
 - storage device, 9-2
- CMOSINIT, performing a, C-10
- codes. *See* error codes
- COL DETECT OFF, Collision Guard instruction, 6-139, 10-145
- COL DETECT ON, Collision Guard instruction, 6-139, 10-145
- cold start, C-12
 - controller, C-7
 - CTRL2, C-6
 - performing a, C-7
- Collision Guard, 10-141
 - false collisions, 10-142
 - instruction, 1-34, 6-4
 - program instructions, COL DETECT OFF, 6-139, 10-145
 - Collision Guard, program instructions, COL DETECT ON, 6-139, 10-145
- command, macro, 3-136
- comment
 - program, 6-7
 - shape generation
 - circle schedule data, 10-55
 - hexagon schedule data, 10-58
 - rectangle schedule data, 10-60
 - slot schedule data, 10-63
- comments
 - analog, I/O, 3-6
 - digital, I/O, 3-12, 3-51, 3-65
 - group, I/O, 3-20
 - group I/O, 3-59
 - UOP, I/O, 3-35
- Communication Protocol, 11-2
 - data format, 11-2
 - flow control mechanism, 11-2
 - hand shaking, 11-4
- communications
 - controller, 1-14, 1-29
 - DeviceNet Interface, 1-29
 - Ethernet, 1-29
 - serial, 1-29
- compatible, IBM-PC, B-1
- complementary
 - output, signals, 3-50
 - output signals, 3-10, 3-63
 - outputs, 3-23
- complementary pairs
 - configuring digital I/O, 3-55
 - output signals, 3-50
- cond, 6-7
- COND Error Codes, A-25
- condition monitor
 - function, 6-135, 10-127
 - instructions, 1-34, 6-4
- condition monitor function, 6-135, 10-127
 - program instructions
 - MONITOR, 6-135
 - MONITOR END, 6-135
 - WHEN, 6-135
- conditional branching, instructions, 6-106
 - IF, 6-106
 - SELECT, 6-106
- configuration, of a position, 6-20

- configuring
 - I/O, 3-5, 3-10, 3-19, 3-33, 3-50, 3-63
 - group, 3-58
 - UOP, I/O, 3-33
 - UOP I/O, 3-58
 - configuring digital I/O, 3-52
 - polarity and complementary pairs, 3-55
 - configuring group I/O, rack, slot, and start point, 3-60
 - Configuring I/O, Rack, Slot, Start Point, 3-40
 - connecting, disk drives, 9-10
 - continuous
 - termination type, 6-33
 - testing, 7-16
 - continuous turn, error recovery limitations, 10-150
 - Control Reliable, MODE SELECT switch, 7-16, 7-26
 - Control Reliable option
 - AUTO STOP, 8-15
 - DEADMAN switch, 1-20, 2-12, 7-14, 7-17
 - General Stop, 8-16
 - MODE SELECT switch, 1-21, 1-23, 7-12, 7-14, 7-16, 7-18, 7-22, 7-23, 7-24, 7-25, 7-27, 7-28
 - robot stop variation, 1-28
 - safety signals
 - Ext E-Stop, 8-15
 - Fence Open (AUTO STOP), 8-15
 - Non Teach Enabling Device (NTED), 8-16
 - Servo Disconnect, 8-16
 - SVON Input, 8-16
 - controlled 2 start, performing a, C-6
 - controlled start, C-12
 - of the controller, C-3
 - performing a, C-4
 - controller, 1-1, 1-2
 - backplane, 1-32
 - capabilities, 1-14
 - cold start, C-7
 - communications, 1-14
 - DeviceNet Interface, 1-29
 - Ethernet, 1-29
 - configuration, 1-16
 - configurations, 1-16
 - controlled start of the, C-3
 - init start, C-2
 - memory, 1-32
 - port configurations
 - optional, 1-29
 - standard, 1-29
 - re-init start, C-10
 - semi hot start, C-9
 - turning off power, 2-4
 - turning on power, 2-3
 - controller backup, 9-1, 9-66
 - controlling
 - digital outputs, 3-51
 - group outputs, 3-59
 - outputs, 3-6, 3-12, 3-20, 3-35
 - controlling I/O, 3-75
 - coordinate system, 2-7
 - Cartesian, 3-78
 - JGFRM, 2-8
 - TOOL, 2-8
 - WORLD, 2-8
 - XYZ, 2-8
 - coordinated, motion option, 6-41
 - coordinated motion, error recovery limitations, 10-150
 - coordinates, 3-78
 - coordinates offset function, 10-110
 - features, 10-111
 - positional data conversion, 10-111
 - axis location, 10-111
 - position and attitude, 10-111
 - robot fixed, 10-113, 10-114
 - rotation speed, 10-111
 - TCP fixed, 10-112, 10-114
 - tool offset, 10-112, 10-113, 10-114
 - screens, 10-110
 - tool offset, 10-110, 10-112, 10-113, 10-114, 10-115
 - convert position data, 10-117
 - convert type, 10-114
 - coordinate system number setting screen, 10-114
 - end line, 10-114
 - insert line, 10-114
 - new program, 10-114
 - new UTOOL number, 10-114
 - old UTOOL number, 10-114
 - original program, 10-114
 - program name setting screen, 10-114
 - range, 10-114
 - start line, 10-114
- UFRAME offset, 10-110, 10-113, 10-117, 10-118
 - coordinate system number setting screen, 10-117
 - end line, 10-117
 - insert line, 10-117
 - new program, 10-117
 - new UTOOL number, 10-117
 - old UTOOL number, 10-117
 - original program, 10-117
 - program name setting screen, 10-117

- range, 10-117
 - start line, 10-117
 - copy source, 6-5
 - copying
 - files to disk, 9-59
 - program files, 9-33
 - program instructions, 5-18
 - corner radii setup, shape generation, hexagon schedule data, 10-58
 - creation date, 6-5
 - cross car mirror, Program ToolBox, 10-184, 10-185, 10-187
 - crt, screen item, C-13
 - CRT/KB, 1-14, 1-26
 - built-in, B-1
 - built-in and external, 1-26
 - keys, B-2
 - menus, B-2
 - remote, B-1
 - setup, B-1, B-2
 - CSTOPI, UOP input signals, 3-35
 - ctrl start, C-12
 - CTRL2 start, C-6
 - current language, setup, 3-152
 - cut direction, shape generation setup, 10-51
 - cut direction override, shape generation setup, 10-51
 - cut shape macros, shape generation, 10-67
 - cycle start, standard operator panel, 7-22, 7-23
- D**
- D-RAM, 1-32
 - data
 - pallet register, 8-11
 - position register, 8-8
 - register, 8-6
 - date
 - of program creation, 6-5
 - of program modification, 6-5
 - DEADMAN switch, Control Reliable option, 1-20, 2-12, 7-14, 7-17
 - deadman switch, 1-20
 - DEC VT-220 terminal, B-1
 - default
 - device settings, 9-5
 - device setup, 9-18
 - defined alarm occurs, manual function detail screen, error recovery, 10-162
 - defined resume program, manual function screen, error recovery, 10-161
 - defining, default program instructions, 5-11
 - deleting
 - files, 9-62
 - program files, 9-34
 - program instructions, 5-17
 - destination position, 6-14
 - detail, of a program, 5-11
 - device
 - default setup, 9-18
 - Memory card, 9-17
 - serial floppy disk, 9-17
 - serial printer, 9-17
 - DeviceNet Interface, 1-29
 - DIAG, C-16
 - using, C-17
 - diag, screen item, C-13
 - diagnostic display. *See* DIAG
 - diameter setup, shape generation
 - circle schedule data, 10-55
 - hexagon schedule data, 10-58
 - DICT Error Codes, A-26
 - digital
 - I/O, 3-3, 3-10, 3-49, 3-50
 - comments, 3-12, 3-51, 3-65
 - rack, 3-10, 3-50, 3-58, 3-63
 - slot, 3-10, 3-51, 3-59, 3-63
 - starting point, 3-10, 3-51, 3-59, 3-63
 - modular I/O board layout, 3-11, 3-20, 3-64
 - process I/O board layout, 3-11, 3-19, 3-64
 - digital I/O
 - configuring, 3-52
 - polarity and complementary pairs, 3-55
 - instructions, 6-100
 - digital input alarms
 - error recovery, 10-156
 - setup, 10-157
 - digital inputs, override select, 3-158
 - DIP switches
 - basic digital I/O module, distributed I/O, 3-47

distributed I/O, 3-46
interface unit, distributed I/O, 3-46
Model B modular I/O
 basic digital I/O unit, 3-47
 interface unit, 3-46
direct, register addressing, 6-93
direct entry method
 jog frame, 3-120, 3-125
 RTCP frame, 3-110
 tool frame, 3-81, 3-90
 user frame, 3-95, 3-105
direct or indirect, register addressing, 6-93
directory, generating a, 9-42
directory of files, 9-41
disk, storage device, 9-2
disk drive
 connecting a, 9-10
 FLOPPY CASSETTE ADAPTER, 9-2
 formatting a, 9-8
 Handy File, 9-2
 PS-100, 9-2
 PS-200, 9-2
 using a, 9-8
displaying
 file contents, 9-58
 I/O, 8-37
 memory status, 8-26
 position status, 8-28
 system variables, 8-13
displaying the alarm log, A-3
distributed I/O, 3-43
 See also Model B modular I/O
 basic digital I/O unit, 3-47
 setup, 3-49
 DIP switches, 3-46
 basic digital I/O unit, 3-47
 example, 3-45
 interface unit, 3-46
 DIP switches, 3-46
 user I/O, 3-49
DNET Error Codes, A-29
DRAM, 1-32
dry run exit/entry, setup item, error recovery, 10-152
dry run speeds, error recovery feature, 10-149
duty diagnosis, 8-38
 duty, 8-39
 duty value, 8-38

 status, 8-39
 temperature, 8-39
duty value
 duty diagnosis, 8-38
 greater than 100%, 8-38
 less than 100%, 8-38
dynamic mastering file, 9-53

E

EDIT key, 5-22, 5-34
editing, background, 5-15, 5-30
editing a program. *See* modifying a program
elements, program, 6-1
ELOG Error Codes, A-36
EMERGENCY STOP button, using the, 7-2
EMON, C-15
 using, C-15
emon, screen item, C-13
\$enable_tcpp, \$TCPPIR structure, 10-179
\$enable_tcpp, \$TCPPIR, 10-179
enabling, brakes, 3-151
ENBL, UOP input signals, 3-35
end effector. *See* applicator
end marker, 6-4
 program, 6-10
entry path, maintenance program, error recovery,
 10-149
error, severity, setting for user alarms, 3-155, 3-157
error code, A-1
 recovery from, A-1
 severity
 ABORT, A-8
 effects of, A-8
 NONE, A-8
 PAUSE, A-7
 SERVO, A-8
 STOP, A-7
 SYSTEM, A-8
 WARN, A-7
 severity descriptions, A-7
error codes
 APSH, A-17
 CD, A-19
 CMCC, A-21
 CMND, A-22

- COND, A-25
- DICT, A-26
- DNET, A-29
- ELOG, A-36
- FILE, A-37
- FLPY, A-40
- FRSY, A-41
- HOST, A-43
- HRTL, A-47
- INTP, A-50
- JOG, A-63
- LANG, A-65
- listing of, A-16
- LNTK, A-66
- MACR, A-69
- MCTL, A-70
- MEMO, A-71
- MOTN, A-75
- PALT, A-94
- PRIO, A-95
- PROG, A-97
- PWD, A-100
- QMGR, A-102
- ROUT, A-103
- SCIO, A-104
- SENS, A-105
- SRVO, A-106
- SYST, A-117, A-118
- TCP, A-122
- TPIF, A-124
- VAR, A-131
- WNDW, A-134
- error log files, backup up, 9-54
- error message
 - contents of, A-3
 - example of, A-3
- error messages
 - severity, user alarm, 3-155, 3-157
 - user alarm, 3-155, 3-157
- error program, instructions, 6-128
- error recovery, 10-146, A-11
 - features, 10-149
 - alarm code monitoring, 10-149
 - alarm recovery screen, 10-149
 - automatic start, 10-149
 - dry run speeds, 10-149
 - error recovery approval DI, 10-149
 - error recovery status DO, 10-149
 - process disable, 10-149
 - program instructions, 10-149
 - test mode, 10-149
 - I/O timing sequence, 10-164
 - limitations, 10-150
 - continuous turn, 10-150
 - coordinated motion, 10-150
 - line tracking, 10-150
 - monitor screen, 10-150
 - multi-tasking systems, 10-150
 - resume program, 10-150
 - single step execution, 10-150
 - soft float, 10-150
 - status line, 10-150
 - maintenance program, 10-146, 10-148, 10-149, 10-159
 - adding instructions, 10-160
 - entry path, 10-149
 - exit path, 10-149
 - maintenance program instructions, 10-159
 - manual function detail screen, 10-162
 - approval DI is on, 10-162
 - automatic error recovery enabled, 10-162
 - defined alarm occurs, 10-162
 - no disabled options, 10-162
 - not in single step mode, 10-162
 - paused and resume program incomplete, 10-162
 - program has motion group, 10-162
 - remote when \$RMT_MASTER is 0, 10-162
 - resume program is defined, 10-162
 - user condition param enabled, 10-162
 - manual function screen, 10-161
 - defined resume program, 10-161
 - detail screen, 10-162
 - error recovery DO status, 10-161
 - operation mode, 10-161
 - manual operation, 10-163
 - normal operation, timing diagram, 10-167
 - normal operation auto start mode, timing diagram, 10-164
 - normal operation when alarm occurs, timing diagram, 10-168
 - overview, 10-146
 - programming, 10-158, 10-160
 - resume program, 10-146, 10-147, 10-149, 10-158
 - adding instructions, 10-160
 - resume program aborted, timing diagram, 10-166
 - resume program instructions, 10-158
 - setup, 10-152, 10-153
 - alarm code monitoring, 10-154
 - approval DI index number item, 10-152
 - automatic start feature item, 10-152
 - digital input alarms, 10-156, 10-157
 - error recovery function item, 10-152
 - incomplete end DO index number item, 10-152
 - items, 10-152
 - maintenance program recovery, 10-152

- reset DI index number item, 10–152
 - resume program recovery, 10–152
 - screen, 10–152
- testing, 10–161
- user alarms, 10–156
- error recovery approval DI, error recovery feature, 10–149
- error recovery DI index number, setup item, error recovery, 10–152
- error recovery DO status, manual function screen, error recovery, 10–161
- error recovery function, setup item, error recovery, 10–152
- error recovery status DO, error recovery feature, 10–149
- errors
 - AUTO mode, 1–25
 - T1 mode, 1–23
 - T2 mode, 1–24
- European controllers
 - AUTO mode, 1–22
 - MODE SELECT switch, 1–22
 - robot stop variation, 1–27
 - T1 mode, 1–22
 - T2 mode, 1–22
- EV, motion option, 6–41
- example
 - direct or indirect, register addressing, 6–93
 - motion option, 6–36, 6–43
 - program, D–1
- example program, 6–3
- executing a macro command, 3–141
- execution history, 8–34
- exit path, maintenance program, error recovery, 10–149
- ext axes only
 - mirror image shift, 10–7
 - shift, 10–20
- Ext E–Stop, safety signal, Control Reliable option, 8–15
- ext integrated
 - mirror image shift, 10–6
 - shift, 10–19
- extended axes, 1–13, 1–31
 - See also* sub-groups
 - mirror image shift, 10–6
 - mirror image shift with ext axes, 10–7

- mirror image shift with ext integrated, 10–6
 - mirror image shift with robot axes only, 10–6
 - shift, 10–19
 - shift with ext axes, 10–20
 - shift with ext integrated, 10–19
 - shift with robot axes only, 10–19
- extended boot monitor. *See* EMON
- extended velocity, motion option, 6–41

F

- FANUC industrialized terminal, B–1
- FANUC Sensor Data Formats, 11–3
- FANUC Sensor Interface, 11–1
- fast exit/entry feature, setup item, error recovery, 10–152
- FAULT, UOP output signals, 3–39
- FAULT RESET, UOP input signals, 3–35
- FCTN
 - key, 1–19
 - menu, 1–19
- FCTN key, C–4
- FCTN menu, 1–36
 - CHANGE RTCP FRAME, 2–13
 - TOGGLE REMOTE TCP, 2–13
- Fence Open, safety signal, Control Reliable option, 8–15
- file
 - backing up from FILE menu, 9–55
 - backing up program, system, and error log files, 9–53
 - copying from FILE menu, 9–59
 - definition of, 9–1
 - displaying contents, 9–58
 - error log files, creating, 9–39
 - loading from FILE menu, 9–45
 - manipulation, 9–1
 - restoring from FILE menu, 9–47
- FILE Error Codes, A–37
- file memory, checking and purging, 9–65
- FILE menu, 9–45, 9–47, 9–55, 9–59
- files
 - application teach pendant program, backing up, 9–53
 - deleting, 9–62
 - error log, backing up, 9–54
 - saving, 9–64

finding, program instructions, 5–20, 5–27

fine, termination type, 6–32

Flash ROM. *See* FROM

flip knuckle, Program ToolBox, 10–184, 10–194

FLOPPY CASSETTE ADAPTER, 9–2

Floppy Cassette adapter, 9–21

floppy disk, 9–17
formatting a, 9–19

FLPY Error Codes, A–40

forcing, I/O, 8–37

forcing outputs, I/O, 3–75

formatting, floppy disks, 9–19

forward, testing, 7–12

four point method, user frame, 3–95

frame
jog, 3–78, 3–79
setup, 3–120
remote TCP, 2–9, 2–13, 3–79, 3–110
remote TCP (RTCP), 3–110
saving, to a file, 3–129
setup, 3–1
tool, 3–78, 3–79
setup, 3–80
user, 3–78
clearing, 3–114
selecting, 3–114, 3–117, 3–119
setup, 3–94
world, 3–78

frame configuration, saving, 3–84, 3–92, 3–99,
3–104, 3–107, 3–114, 3–118, 3–124, 3–127,
3–130

frame setup, 9–63

frames
how used, 3–78
kinds of, 3–78
setup, 3–78
shape generation, 10–75

FROM
amount of, 1–32
memory, 1–32

from, screen item, C–13

FRSY Error Codes, A–41

full menus, 1–19, 1–36

G

GEFanuc Genius I/O, network, 3–4, 3–49

General Stop, Control Reliable option, 8–16

generating a, directory, 9–42

group, I/O, 3–3, 3–18, 3–49, 3–58
comments, 3–20
number of points, 3–19
rack, 3–19
slot, 3–19
starting point, 3–19

group I/O
comments, 3–59
configuring, 3–58
rack, slot, and start point, 3–60
instructions, 6–104
number of points, 3–59
simulating, 3–59

group mask, motion group, 6–8

group outputs, controlling, 3–59

\$group_num, \$TCPP_CFG structure, 10–183

\$group_num, \$TCPP_CFG, 10–183

groups, motion, 2–10

guidelines, for programming, 5–2

H

hand breakage recovery, A–13

HandlingTool
program, 1–33
setup, 1–33
system, 1–2

HandlingTool Software, software, 1–33

HandlingTool software, 1–1

Handy File, 9–23

Handy File disk drive, 9–2

hardware, Auto TCP, 12–1, 12–3

HELD, UOP output signals, 3–39

hexagon schedule data, shape generation
blend-in angle setup, 10–58
blend-in distance setup, 10–58
comment, 10–58
corner radii setup, 10–58
diameter setup, 10–58
overlap angle setup, 10–58
schedule number setup, 10–58

- speed setup, 10-58
- start axis setup, 10-58
- x-work angle setup, 10-58
- y-work angle setup, 10-58
- hexagon schedules, 10-57
- hexagons, shape generation, 10-50, 10-53
- hints, mirror image, 10-11
- HOLD, UOP input signals, 3-35
- HOLD button, using the, 7-3
- HOME, UOP input signals, 3-35
- home position, perch, 5-6
- HOST Error Codes, A-43
- HRTL Error Codes, A-47
- I**
- I/O, 1-30
 - AI, 1-30
 - Allen Bradley Remote, 3-4, 3-49
 - analog, 3-3, 3-4
 - comments, 3-6
 - instructions, 6-103
 - AO, 1-30
 - channel, analog, 3-5
 - configuring, 3-5, 3-10, 3-19, 3-33, 3-50, 3-63
 - group, 3-58
 - controlling, 3-75
 - DI, 1-30
 - digital, 3-3, 3-10, 3-49, 3-50
 - comments, 3-12, 3-51, 3-65
 - instructions, 6-100
 - displaying, 8-37
 - distributed, 3-43
 - DO, 1-30
 - Ethernet, 1-30
 - forcing, 8-37
 - forcing outputs, 3-75
 - GEFanuc Genius, 3-4, 3-49
 - Genius I/O, network, 1-30
 - GI, 1-30
 - GO, 1-30
 - group, 3-3, 3-18, 3-49, 3-58
 - comments, 3-20, 3-59
 - instructions, 6-104
 - instructions, 1-33, 6-3, 6-100
 - interconnect, setup, 3-29
 - Link screen, 3-1
 - Model A modular, setup, 3-43
 - modular, Model B, 3-43, 3-44, 3-46
 - modular board layout
 - analog, 3-6
 - digital, 3-11, 3-20, 3-64
 - UOP, 3-34
 - Modular I/O, 1-30
 - Modular I/O layout, 3-4
 - number of points, group, 3-19, 3-59
 - override select, 3-158
 - plc, 3-3, 3-49
 - PLC Inputs, 1-30
 - PLC Outputs, 1-30
 - polarity, 3-50
 - process, setup, 3-43
 - process board layout, 3-3
 - analog, 3-5
 - digital, 3-11, 3-19, 3-64
 - UOP, 3-34
 - Process I/O, Standard, 1-30
 - quantity of, 1-30
 - rack
 - analog, 3-5
 - digital, 3-10, 3-50, 3-58, 3-63
 - group, 3-19
 - UOP, 3-33
 - Remote I/O, 1-30
 - remote interfaces, 1-30
 - RI, 1-30
 - RO, 1-30
 - robot, 3-3, 3-23, 3-49
 - instructions, 6-101
 - saving, 3-54, 3-57, 3-62
 - saving information, 3-42, 3-69
 - sensor I/O signal cables, Auto TCP, 12-5
 - setup, 3-3, 3-43
 - Model B modular I/O example, 3-45
 - simulating, 3-6, 3-12, 3-20, 3-51, 3-76
 - slot
 - analog, 3-5
 - digital, 3-10, 3-51, 3-59, 3-63
 - group, 3-19
 - UOP, 3-33
 - SOP, 3-3, 3-49, 8-36
 - SI, 1-30
 - SO, 1-30
 - starting point
 - digital, 3-10, 3-51, 3-59, 3-63
 - group, 3-19
 - UOP, 3-33
 - UOP, 3-3, 3-33
 - comments, 3-35
 - UI, 1-30
 - UO, 1-30
 - user, 3-3, 3-49
 - VOP, 3-49

- I/O configuration, 9-63
- I/O interconnect, SI to DO, 3-28, 3-31
- I/O interconnect
 - ES to DO, 3-28, 3-32
 - MODE SELECT switch, 3-31
- I/O Link screen, 3-1
 - Model B I/O, 3-1
- I/O setup, interconnect, 3-29
- I/O timing sequence, error recovery, 10-164
- IBM-PC compatible, B-1
- IF
 - conditional branching instructions, 6-106
 - I/O, 6-107
- if, instruction, 6-106
- ignore pause, 6-9
- IMSTP, UOP input signals, 3-35
- incomplete DO index number, setup item, error recovery, 10-152
- incremental, motion option, 6-40
- indicators, 8-2
 - operator panel, 8-3
 - teach pendant, 8-2
 - BUSY, 8-2
 - FAULT, 8-2
 - GUN ENBL, 8-2
 - HOLD, 8-2
 - I/O ENBL, 8-2
 - JOINT, 8-2
 - RUNNING, 8-2
 - STEP, 8-2
 - TOOL, 8-2
 - WELD ENBL, 8-2
 - XYZ, 8-2
- indirect, register addressing, 6-93
- init start, C-12
 - controller, C-2
- initializing, ports, 9-3
- input, robot, 3-49
- input signals, macro commands, 3-138
- input/output. *See* I/O
- inputs, UOP, 3-35
- inputs and outputs, simulating, 3-76
- inserting, program instructions, 5-17, 5-24
- install, screen item, C-13
- install option, screen item, C-4
- install update, screen item, C-4
- installation, Auto TCP hardware, 12-1, 12-3
- installing, shape generation, 10-50
- instruction
 - arc start, 10-158
 - if, 6-106
 - jump, 6-105
 - maximum speed, 6-118
 - message, 6-113
 - motion group, 6-131
 - offset condition, 6-37
 - override, 6-113
 - parameter name, 6-114
 - remark, 6-113
 - RSR enable/disable, 6-112
 - run program, 6-125
 - select, 6-108
 - sensor, 6-129
 - skip, 6-119
 - SKIP CONDITION, 6-36
 - subprogram call, 6-106
 - timer, 6-113
 - user alarm, 6-112
- instructions
 - abort, 6-127
 - adjusting programs during program or production run, 7-30
 - analog I/O, 6-103
 - backing up system files, TPP programs, and application files on disk, 9-55
 - branching, 1-33, 6-3, 6-105
 - CLEAR_PATH_DSBL, 10-159
 - CLEAR_RESUME_PROG, 10-158
 - Collision Guard, 1-34, 6-4
 - COL DETECT OFF, 6-139, 10-145
 - COL DETECT ON, 6-139, 10-145
 - condition monitor, 1-34, 6-4
 - MONITOR, 6-135
 - MONITOR END, 6-135
 - WHEN, 6-135
 - conditional branching, 6-106
 - IF, 6-106
 - SELECT, 6-106
 - configuring PLC I/O
 - polarity and complementary pairs, 3-67
 - rack, slot, start point, 3-65
 - configuring UOP I/O, rack, slot, start point, 3-40
 - continuous testing
 - using operator panel, 7-18
 - using teach pendant, 7-16
 - copying a program within the SELECT Menu, 9-33

- create and write a new program, 5-12
 - deleting programs from the SELECT Menu, 9-34
 - digital I/O, 6-100
 - displaying and setting position registers, 8-8
 - displaying and setting registers, 8-6
 - displaying and setting system variables, 8-13
 - displaying memory status, 8-26
 - displaying position status, 8-28
 - displaying the contents of a Text (ASCII) File, 9-58
 - displaying the user screen, 8-5
 - displaying the version identification status, 8-23
 - EMERGENCY STOP, 7-2
 - error program, 6-128
 - executing a macro command, from Manual FCTNS, 3-144
 - formatting a disk, 9-19
 - generating a directory of files, 9-42, 9-65
 - group I/O, 6-104
 - HOLD and recovery, 7-3
 - I/O, 1-33, 6-3, 6-100
 - inserting a line, 6-15
 - label definition, 6-105
 - loading a program, 9-31
 - loading files using the FILE Menu, 9-45
 - macro, 1-34, 6-4
 - macro command, 6-126
 - MAINT_PROG, 10-159
 - maintenance program, adding, 10-160
 - maintenance program, 10-159
 - miscellaneous, 1-34, 6-4, 6-112
 - monitoring a running program, 7-20
 - motion, 1-33, 6-3, 6-11
 - multiple control, 1-34, 6-4, 6-125
 - offset, 1-34, 6-4, 6-121
 - offset condition, 6-121
 - pause, 6-127
 - payload, 1-34, 6-4, 6-137
 - position register, 1-33, 6-3, 6-96
 - position register element, 6-97
 - position register look-ahead, 1-34, 6-4, 6-134
 - position register look-ahead function
 - LOCK PREG, 6-134
 - UNLOCK PREG, 6-134
 - PR[i,j], 6-97
 - PR[x], 6-96
 - printing a program, 9-36
 - printing a teach pendant screen, 9-38
 - program, 6-3
 - error recovery, 10-149
 - program control, 1-34, 6-4, 6-127
 - recovery from EMERGENCY STOP, 7-3
 - register, 1-33, 6-3, 6-91, 6-93
 - resume program, 6-128, 10-158
 - adding, 10-160
 - RESUME_PROG, 10-158
 - robot I/O, 6-101
 - saving a program to a disk, 9-29
 - saving files to the default device, 9-64
 - saving frame data to a file, 3-129
 - saving I/O information, 3-42, 3-69
 - selecting a jog frame, 3-128
 - selecting a program on the select menu, 9-28
 - selecting a tool frame, 3-93
 - selecting user frame, 3-108, 3-119
 - semaphore, 6-125
 - sensor, 1-34, 6-4
 - setting brake on hold, 3-151, 3-178
 - setting brake timers, 3-148
 - setting the default device, 9-18
 - setting up a port, 9-7
 - setting up axis limits, 3-146
 - setting up frame using direct entry method, 3-90
 - setting up six point method, 3-85
 - setting up the jog frame using the direct entry method, 3-125
 - setting up the jog frame using the three point method, 3-121
 - setting up the user frame using the three point method, 3-95, 3-111
 - setting up three point method, 3-82
 - setting up user frame using the direct entry method, 3-105
 - setup macro command, 3-139
 - simulating inputs and outputs, I/O, 3-76
 - single step testing, 7-14
 - skip, 1-33, 6-4
 - uframe, 6-123
 - uframe_num, 6-122
 - unconditional branching, 6-105
 - jump, 6-105
 - subprogram call, 6-105
 - using a floppy disk and disk drive, 9-10
 - using mirror image, 10-7
 - using the shift utility, 10-21
 - using wait release, 7-21
 - utool, 6-123
 - utool_num, 6-122
 - wait, 1-33, 6-4, 6-109
 - wait condition, 6-109
 - wait time, 6-109
 - wait condition, 6-109
 - forever, 6-109
 - timeout, LBL[1], 6-109
 - wait semaphore, 6-125
 - wait time, 6-109
- IntelliTrak, 6-58

interconnect I/O
 menu, 3-3, 3-29
 setup, 3-29

interface
 formatting a, memory card, 9-11
 memory card, 9-11
 using a memory card, 9-11

interface unit
 DIP switches, distributed I/O, 3-46
 Model B modular I/O, configuring DIP switches,
 3-46

INTP Error Codes, A-50

inverse polarity, 3-50

instructions, modifying a program, 5-22, 5-33

J

JGFRM, coordinate system, 2-8

JOG Error Codes, A-63

jog frame, 3-78, 3-79
 selecting, 3-128
 setup, 3-120
 direct entry method, 3-120, 3-125
 three point method, 3-120, 3-121

jog menu, 2-15

jog speed, 2-6

jogging, 2-11, 5-1
 best frame used in, 5-2
 jog menu, 2-15
 remote TCP, 2-9, 2-13
 the robot, 2-1, 2-5

jogging the robot
 and other axes, 2-11
 changing group, 2-12
 COORD display, 2-7
 coordinate system, 2-5, 2-7
 de-select sub-group, 2-12
 extended axes and motion sub-groups, 2-5
 extended axes and sub-groups, 2-10
 jog speed delay, 2-6, 2-10
 jog speed keys, 2-6
 jog speed values, 2-6
 minor axis wrist jog, 2-5
 motion groups, 2-5
 select a coordinate system, 2-11
 select a jog speed, 2-13
 select sub-group, 2-12
 select wrist jog, 2-12

to jog, 2-14
 using wrist jog, 2-9
 when finished jogging, 2-14

Joint, Definition of, 1-3

joint
 coordinate system, 2-7
 motion, 1-31
 motion type, 6-12
 position status, 8-28

joint motion, speed, 6-23

JOINT_MAX_SPEED, program instruction, 6-118

jump
 branching instructions, 6-105
 instruction, 6-105

K

kerf width, shape generation setup, 10-51

kerf width override, shape generation setup, 10-51

keys, CRT/KB, B-2

knuckle, flip, Program ToolBox, 10-184, 10-194

L

label definition, instructions, 6-105

LANG Error Codes, A-65

\$lc_qstp_enb, \$PARAM_GROUP[1], 1-25

\$lc_qstp_enb, \$PARAM_GROUP[1] structure, 1-25

LEDs. *See* indicators

length setup, shape generation
 rectangle schedule data, 10-60
 slot schedule data, 10-63

limit, set, Program ToolBox, 10-184, 10-197

limit set, Program ToolBox, 10-184, 10-197

limitations, Torch Guard, 10-141

limits, axis, setup, 3-145

line number, 6-3, 6-10

line tracking, error recovery limitations, 10-150

linear
 motion, 1-31
 motion type, 6-13

linear motion, speed, 6-24

LINEAR_MAX_SPEED, program instruction, 6-118

LNTK Error Codes, A-66

load files from disk to controller, 9-44

loadable files, 9-44

loading

- files from FILE menu, 9-45
- programs, 9-31

location, 3-78

- moving a frame's, 3-79
- of a position, 6-20

LOCK PREG, position register look-ahead function instruction, 6-134

locking

- AUTO mode, 1-25
- T1 mode, 1-23
- T2 mode, 1-24

look-ahead function, for position registers, 6-134

lower limits, axes, 3-146

M

M-16i, 1-5

M-400, 1-6

M-410i, 1-6

M-500, 1-7

M-6i, 1-5

MACR Error Codes, A-69

macro, 6-7

macro command, 3-136

- executing a, 3-141
- executing from MAN FCTNS, 3-144
- executing from the teach pendant, 3-141
- input signals, 3-138
- instructions, 1-34, 6-4, 6-126
- running with MAN FCTNS screen, 3-137
- setting up, 3-139
- setting up a, 3-136

macro setup, 9-63

macros, 5-4

MAINT DO index number, setup item, error recovery, 10-152

MAINT_PROG, instruction, 10-159

maintenance program

- error recovery, 10-146, 10-148, 10-149, 10-159
- entry path, 10-149
- exit path, 10-149
- instructions, adding, 10-160

- setup, 10-152
 - dry run exit/entry item, 10-152
 - fast exit/entry feature item, 10-152
 - MAINT DO index number item, 10-152
 - maintenance program item, 10-152
 - setup item, error recovery, 10-152

Major Axes, Picture of, 1-3

MAN FCTNS, 3-137

- executing macro commands from, 3-144

Manipulating Files

- displaying text (ASCII) files, 9-58
- file types, 9-40
 - command file, 9-40
 - data file, 9-40
 - default file, 9-40
 - I/O file, 9-40
 - KAREL file, 9-40
 - listing file, 9-40
 - mn file, 9-40
 - P-Code file, 9-40
 - system file, 9-40
 - text file, 9-40
 - variable file, 9-40
- generating a sub-directory, 9-41
- saving files, 9-63

manipulating files, 9-39

- backing up program, system, and files, 9-53

manual data entry, BMON, C-14

manual function detail screen, error recovery, 10-162

- approval DI is on, 10-162
- automatic error recovery enabled, 10-162
- defined alarm occurs, 10-162
- no disabled options, 10-162
- not in single step mode, 10-162
- paused and resume program incomplete, 10-162
- program has motion group, 10-162
- remote when \$RMT_MASTER is 0, 10-162
- resume program is defined, 10-162
- user condition param enabled, 10-162

manual function screen, error recovery, 10-161

- defined resume program, 10-161
- detail screen, 10-162
- error recovery DO status, 10-161
- operation mode, 10-161

manual operation, error recovery, 10-163

MCTL Error Codes, A-70

mastering methods

- fixture position master, E-1
- quick mastering, E-1
- single axis mastering, E-1
- zero degree mastering, E-1

- maximum speed, instruction, 6-118
- mcard, screen item, C-13
- MEMO Error Codes, A-71
- memory
 - checking and purging file, 9-65
 - CMOS RAM, 1-32
 - controller, 1-32
 - D-RAM, 1-32
 - DRAM, 1-32
 - Flash ROM, 1-32
 - status, 8-26
 - hardware, 8-26
 - system, 8-26
 - user, 8-26
- Memory card, 9-17
- memory card, interface, 9-11
- memory card interface, connecting a, 9-11
- menu
 - full, 1-36
 - interconnect I/O, 3-3, 3-29
 - jog, 2-15
- menu maps, 1-1
- menus
 - CRT/KB, B-2
 - full, 1-19
 - quick, 1-19
- MENUS key, C-4
- message, instruction, 6-113
- mirror, cross car, Program ToolBox, 10-184, 10-185, 10-187
- mirror image, 10-4
 - hints, 10-11
 - parallel, 10-4
 - replace ext axes, 10-7
 - rotational, 10-4, 10-5
 - shift ext axes only, 10-7
 - shift with ext axes, 10-7
 - shift with ext integrated, 10-6
 - shift with robot axes only, 10-6
- mirror image program shift, 10-1
- mirror image shift
 - extended axes, 10-6
 - using the, 10-7
- miscellaneous, instructions, 1-34, 6-4, 6-112
- mode, test, error recovery, 10-149
- MODE SELECT switch, 3-31
 - AUTO, 7-26
 - AUTO mode, 1-24, 7-18, 7-22, 7-23, 7-24, 7-25, 7-27, 7-28
 - Control Reliable, 7-16, 7-26
 - Control Reliable option, 1-21, 1-23, 7-12, 7-14, 7-16, 7-18, 7-22, 7-23, 7-24, 7-25, 7-27, 7-28
 - European controllers
 - AUTO mode, 1-22
 - T1 mode, 1-22
 - T2 mode, 1-22
 - for European controllers, 1-22
 - T1, 7-16
 - T1 mode, 1-23, 7-12, 7-14, 7-16
 - T2, 7-16
 - T2 mode, 1-24, 7-12, 7-14, 7-16
- Model B I/O, I/O Link screen, 3-1
- Model B modular I/O, 3-43, 3-44
 - configuring DIP switches
 - on a basic digital I/O unit, 3-47
 - on the interface unit, 3-46
 - example, 3-45
 - hardware configuration, 3-46
 - setup tasks, 3-43
- modification date, 6-5
- modifying
 - a program, 5-1
 - other instructions, 5-17, 5-24
- modifying a program, 5-17, 5-22, 5-33
 - in the background, 5-30
- modular, I/O board layout
 - analog, 3-6
 - digital, 3-11, 3-20, 3-64
 - UOP, 3-34
- Modular I/O, I/O board layout, 3-4
- modular I/O, Model A, setup, 3-43
- MONITOR, condition monitor instructions, 6-135
- monitor a program, 7-20
- MONITOR END, condition monitor instructions, 6-135
- monitoring, alarm code, error recovery feature, 10-149
- motion, 1-31
 - circular, 1-31
 - groups of, 1-31
 - instructions, 1-33, 6-11
 - joint, 1-31
 - linear, 1-31

minimizing changes in wrist orientation, 5-3
 robot, 1-14
 types of, 1-31
 motion development, screen item, C-3
 motion group, instruction, 6-131
 motion groups, 1-31, 2-10, 6-8
 motion instruction, example, 6-11
 motion instructions, 6-3
 motion option
 EV, 6-41
 Simultaneous EV, 6-41
 motion options, 6-11, 6-34
 acceleration override, 6-35
 coordinated, 6-41
 EV, 6-41
 extended velocity, 6-41
 incremental, 6-40
 offset, 6-37
 Offset, PR[x], 6-37
 PTH, 6-44
 remote TCP, 6-42
 skip label, 6-36
 Skip, LBL[x], 6-36
 time before/after, 12-1
 Tool_offset, 6-38, 6-124
 Tool_offset, PR[], 6-39
 wrist joint, 6-34
 motion range. *See* axis limits
 motion sysvar setup, screen item, C-3
 motion type, 6-11, 6-12
 circular, 6-14, 6-15
 circular orientation control, at intermediate via
 point, 6-15
 joint, 6-12
 linear, 6-13
 MOTN Error Codes, A-75
 motor, duty diagnosis, 8-38
 \$motype, \$TCPPSPEED structure, 10-180
 \$motype, \$TCPPSPEED, 10-180
 moving
 the location of a frame, 3-79
 the orientation of a frame, 3-79
 moving the robot. *See* jogging
 multi-tasking, 10-30
 multiple control, instructions, 1-34, 6-4, 6-125

N

naming a program, 5-10, 6-6
 no disabled options, manual function detail screen,
 error recovery, 10-162
 noload, screen item, C-12
 Non Teach Enabling Device (NTED), safety signal,
 Control Reliable option, 8-16
 normal operation, timing diagram, error recovery,
 10-167
 normal operation auto start mode, timing diagram,
 error recovery, 10-164
 normal operation when alarm occurs, timing diagram,
 error recovery, 10-168
 normal operation without resume program execution,
 timing diagram, error recovery, 10-165
 normal polarity, 3-50
 not in single step mode, manual function detail screen,
 error recovery, 10-162
 NTED, Non Teacher Enabling Device, Control
 Reliable option, 8-16
 number of points, I/O, group, 3-19, 3-59
 number of positions, 6-6

O

offset
 coordinates, 10-110
 instructions, 1-34, 6-4, 6-121
 motion option, 6-37
 tool, coordinates offset function, 10-110, 10-112,
 10-113, 10-114, 10-115
 uframe, coordinates offset function, 10-110,
 10-113, 10-117, 10-118
 offset condition
 instruction, 6-37
 instructions, 6-121
 Offset, PR[x], motion option, 6-37
 \$oft_tim_enb, \$TCPP_CFG structure, 10-183
 \$oft_tim_enb, \$TCPP_CFG, 10-183
 OLPC, 9-3
 operation mode, manual function screen, error
 recovery, 10-161
 operations, programming, undoing, 5-20, 5-28
 operator panel
 indicators, 8-3
 BATTERY ALARM, 8-3

- FAULT, 8-3
- REMOTE, 8-3
- TEACH PENDNAT ENABLED, 8-3
- USER LED #1, 8-3
- USER LED #2, 8-3
- signal setup
 - axis limits, 3-2
 - brake on hold, 3-2
 - brake timers, 3-2
 - UOP, 3-1
- standard, 1-20
- user, 1-26

- option package, PLC I/O, 3-63

- orient origin point, 3-87

- orientation, 3-78
 - moving a frame's, 3-79
 - of a position, 6-20

- origin, 3-78

- output, robot, 3-49

- outputs
 - complementary, 3-23
 - complementary signals, 3-10, 3-50, 3-63
 - controlling, 3-6, 3-12, 3-20, 3-35
 - digital, controlling, 3-51
 - group, controlling, 3-59
 - UOP, 3-39

- overlap angle setup, shape generation
 - circle schedule data, 10-56
 - hexagon schedule data, 10-58
 - rectangle schedule data, 10-61
 - slot schedule data, 10-64

- override, instruction, 6-113

- override registers, use in shape generation, 10-69

- override select, digital inputs, 3-158

- override select, setup, 3-158, 3-160

- overtravel recovery, A-11

- overview, Auto TCP, 12-1, 12-2

- P**
 - P2. *See* P3 port
 - P2 port, 9-8
 - pairs
 - complementary, output signals, 3-50
 - configuring digital I/O, 3-55
 - pallet register, data, 8-11
 - PALT Error Codes, A-94
 - Panel Digital Inputs. *See* PDI
 - Panel Digital Outputs. *See* PDO
 - parallel, mirror image, 10-4
 - parallel shift, 10-16, 10-17, 10-18
 - \$PARAM_GROUP[1]
 - .\$lc_qstp_enb, 1-25
 - .\$t1t2_sngstp, 1-23, 1-24, 1-25, 7-17
 - \$PARAM_GROUP[1].\$lc_qstp_enb, 1-25
 - \$PARAM_GROUP[1].\$t1t2_sngstp, 1-23, 1-24, 1-25, 7-17
 - parameter name, instruction, 6-114
 - pasting, program instructions, 5-18
 - in reverse, 5-18
 - PATH Switch, motion option, pth switch, 6-44
 - pause, instructions, 6-127
 - PAUSED, UOP output signals, 3-39
 - paused and resume program incomplete, manual function detail screen, error recovery, 10-162
 - payload
 - instruction, 1-34, 6-4
 - instructions, 6-137
 - setup, 3-176
 - PC compatible, disk drive, 9-3
 - PCB, axis control, 3-49
 - PDI, 8-36
 - PDO, 8-36
 - personal computer, storage device, 9-3
 - planning
 - a program, 5-1
 - program, 5-2
 - plc, I/O, 3-3, 3-49
 - PLC I/O, option package, 3-63
 - PNS, 1-30
 - See also* program number select (PNS)

- point sets, Auto TCP, 12-13
 - manually defining, 12-14
- polarity, 3-10, 3-23, 3-50
 - configuring digital I/O, 3-55
 - PLC I/O setup, 3-63
- port
 - default device, 9-5
 - initializing, 9-3
 - setting up a, 9-7
- ports, 9-5
- position, 3-78
 - destination, 6-14
 - positional data conversion, coordinates offset function, 10-111
 - status, 8-28
 - via, 6-14
- position register, 5-5
 - data, 8-8
 - instructions, 1-33, 6-3, 6-96
 - look-ahead, 1-34, 6-4, 6-134
- position register element, instructions, 6-97
- position register look-ahead function, 6-134
 - program instructions
 - LOCK PREG, 6-134
 - UNLOCK PREG, 6-134
- position registers, 9-63
 - look-ahead function, 6-134
- positional data, conversion, coordinates offset function, 10-111
- positional information, 6-11, 6-20
- power
 - turning off controller, 2-4
 - turning on controller, 2-3
- PR[i,j], instructions, 6-97
- PR[x], instructions, 6-96
- predefined positions, 5-4
 - macros, 5-4
 - other, 5-8
 - position registers, 5-5
 - programs, 5-4
 - safe position, pounce, 5-8
- printer, 9-17
 - requirements for, 9-35
- printing
 - programs, 9-35, 9-36
 - requirements for, 9-35
 - teach pendant screens, 9-38
- PRIO Error Codes, A-95
- process, I/O board layout, 3-3
 - analog, 3-5
 - digital, 3-11, 3-19, 3-64
 - UOP, 3-34
- process disable, error recovery feature, 10-149
- process I/O, setup, 3-43
- process macros, shape generation, 10-68
- process off macro, shape generation, 10-68
- process on macro, shape generation, 10-68
- production, 7-1, 7-22
 - running, 1-35
 - setup, 3-1
- production mode, shape generation, 10-70
- production operation, 3-131
 - program adjust, 7-29, 7-30
 - program number select (PNS), 3-134, 7-27, 10-33
 - robot service request (RSR), 3-131, 7-25, 7-26
 - standard operator panel cycle start, 7-22, 7-23
 - UOP production start, 7-27, 10-33
 - user operator panel start, 7-24
- PROG Error Codes, A-97
- prog init, screen item, C-3
- \$prog_speed, \$TCPPSPEED structure, 10-180
- \$prog_speed, \$TCPPSPEED, 10-180
- program, 1-33
 - application teach pendant files, backing up, 9-53
 - background program editing, 5-15, 5-30
 - default instructions, 5-11
 - detail, 5-11
 - end marker, 6-10
 - example, 1-34, 6-3
 - examples, D-1
 - execution history, 8-34
 - header, 6-3
 - ignore pause, 6-9
 - instructions, 6-3
 - macro, 3-136
 - maintenance, error recovery, 10-146, 10-148, 10-149
 - manipulation, 9-1
 - mirroring portions of a, 10-10
 - mirroring positions in an entire, 10-9
 - modifying a, 5-1, 5-9, 5-17
 - modifying in the background, 5-30
 - naming a, 5-10
 - planning, 5-1, 5-2
 - planning a, 5-1

- printing a, 9-35
- protection, 6-8
- resume, error recovery, 10-146, 10-147, 10-149
- selecting a, 5-17
- shift utility, 10-16
- speed, 6-23
- sub type, 6-7
 - ch, 6-7
- testing, 7-1, 7-9
- testing a, 1-35
- touching up a, 5-17, 5-22
- writing a, 5-1, 5-9, 5-10
- program activation
 - AUTO mode, 1-24
 - T1 mode, 1-23, 1-24
- program adjust, 7-29, 7-30
 - guidelines, 7-29
 - schedules, 7-29
- Program and File Manipulation
 - a program, 9-1
 - files, 9-1
 - storage devices, 9-1
- program base shift, 10-1
- program comment, 6-7
 - available characters for, 6-7
 - content of, 6-7
 - length of, 6-7
- program control, instructions, 1-34, 6-4, 6-127
- program elements, 6-1
- program end, 6-4
- program examples, D-1
 - conditional branching using labels, D-3
 - group output, D-4
 - jump label, D-5
 - labels, D-5
 - macro instruction, D-6
 - message, D-5
 - position register value, D-2
 - pulse instruction, D-4
 - register analog input, D-3
 - register increment, D-4
 - wait instruction, D-4
- program file, backing up, 9-55
- program files
 - copying, 9-33
 - deleting, 9-34
 - saving, 9-29
 - selecting, 9-28
- program has motion group, manual function detail screen, error recovery, 10-162
- program header, 6-5
- program instructions, error recovery feature, 10-149
- program name, 6-6
 - available characters for, 6-6
 - content of, 6-6
 - length of, 6-6
- program number select (PNS), 3-134, 7-27, 10-33
 - sequence, 3-134
 - setting up, 3-135
 - setup items, 3-135
- program planning
 - motion, 5-2
 - predefined positions, 5-4
 - home position (perch), 5-6
 - macros, 5-4
 - other positions, 5-8
 - position registers, 5-5
 - programs, 5-4
 - repair position, 5-7
- program shift, mirror image, 10-1
- program size, 6-6
- program timer, status, 8-18
- Program ToolBox, 10-3, 10-184
 - cross car mirror, 10-184, 10-185, 10-187
 - flip knuckle, 10-184, 10-194
 - limit set, 10-184, 10-197
 - setup, 10-3, 10-184
 - UTOOL adjust, 10-184, 10-189
- programming
 - adding instructions, 5-11
 - copying instructions, 5-18
 - deleting instructions, 5-17
 - error recovery, 10-158, 10-160
 - finding instructions, 5-20, 5-27
 - guidelines, 5-2
 - hints, 5-2
 - inserting instructions, 5-17, 5-24
 - modifying a program, 5-22, 5-33
 - modifying other instructions, 5-17, 5-24
 - monitor a program, 7-20
 - pasting instructions, 5-18
 - renumbering positions, 5-20, 5-28
 - replacing instructions, 5-20, 5-27
 - shape generation, 10-65
 - undoing operations, 5-20, 5-28
- programs
 - loading, 9-31
 - loading from disk, 9-31

PROGRUN, UOP output signals, 3-39

PS-100 disk drive, 9-2

PS-200 disk drive, 9-2

PTH, motion option, 6-44

purging, file memory, 9-65

PWD Error Codes, A-100

\$PWF_IO, 4-2

\$PWR_NORMAL, 4-2

Q

QMGR Error Codes, A-102

quick menus, 1-19

R

R-J2 controller. *See* controller

rack, I/O

analog, 3-5

digital, 3-10, 3-50, 3-58, 3-63

group, 3-19

UOP, 3-33

rack, slot, and start point, configuring group I/O, 3-60

radius 1 setup, shape generation, rectangle schedule data, 10-61

radius 2 setup, shape generation, rectangle schedule data, 10-61

radius 3 setup, shape generation, rectangle schedule data, 10-61

radius 4 setup, shape generation, rectangle schedule data, 10-61

RAM. *See* CMOS RAM

range, motion. *See* axis limits

recovering from a hand breakage, A-13

recovering from an overtravel, A-11

recovery, 7-2, 7-3

error, 10-146

from errors, A-11

hand breakage, A-13

resume tolerance, 7-4

rectangle schedule data, shape generation

blend-in angle setup, 10-61

blend-in distance setup, 10-61

comment, 10-60

length setup, 10-60

overlap angle setup, 10-61

radius 1 setup, 10-61

radius 2 setup, 10-61

radius 3 setup, 10-61

radius 4 setup, 10-61

schedule number setup, 10-60

speed setup, 10-61

start axis setup, 10-61

width setup, 10-60

x-work angle setup, 10-61

y-work angle setup, 10-61

rectangles, shape generation, 10-50, 10-53

reference frame, 3-78

reference position, 10-43

register, 9-63

addressing, direct or indirect, 6-93

data, 8-6

instructions, 1-33, 6-3, 6-91, 6-93

re-init start, controller, C-10

reinit start, using CMOSINIT, C-10

releasing, program execution, 7-21

remark, instruction, 6-113

remarks, 6-4

remote

CRT/KB, B-1

I/O interfaces, 1-30

remote TCP

CHANGE RTCP FRAME, 2-13

coordinate system display, 2-9

frame, 2-9, 2-13, 3-79, 3-110

jogging, 2-9, 2-13

motion option, 6-42

TOGGLE REMOTE TCP, 2-13

user frame, 2-9, 2-13, 3-110

remote TCP frame, 3-79, 3-110

remote when \$RMT_MASTER is 0, manual function detail screen, error recovery, 10-162

renumbering positions, 5-20, 5-28

repair position, 5-7

replace ext axes

mirror image shift, 10-7

shift, 10-21

replacing, program instructions, 5-20, 5-27

requirements, Auto TCP, 12-1, 12-3

reset DI index number, setup item, error recovery, 10-152

- restoring
 - a controller, 9-66
 - files from FILE menu, 9-47
- restoring a, controller, 9-72
- resume, tolerance, 7-4
- resume program
 - defined, on manual function screen for error recovery, 10-161
 - error recovery, 10-146, 10-147, 10-149, 10-158
 - instructions, 6-128
 - adding, 10-160
 - setup, 10-152
 - auto start max count item, 10-152
 - auto start max count R[] item, 10-152
 - status DO index number item, 10-152
- resume program aborted, timing diagram, error recovery, 10-166
- resume program is defined, manual function detail screen, error recovery, 10-162
- RESUME_PROG, instruction, 10-158
- RETURN_PATH_DSBL, instruction, 10-159
- RI/RO signals, axis control board, 3-49
- RIO, PLC I/O setup, 3-63
- \$RMT_MASTER, error recovery, 10-162
- robot, 1-1
 - A-520i, 1-4
 - fixed, positional data conversion for coordinates offset function, 10-113, 10-114
 - I/O, 3-3, 3-23, 3-49
 - M-16i, 1-5
 - M-400, 1-6
 - M-410i, 1-6
 - M-500, 1-7
 - M-6i, 1-5
 - M-710i, 1-7
 - motion, 1-14
 - S-12, 1-8
 - S-420, 1-9
 - S-420iW, 1-9
 - S-450, 1-10
 - S-500, 1-10
 - S-6, 1-8
 - S-700, 1-11
 - S-800, 1-11
 - S-900W, 1-12
 - setting payload, 3-176
 - turning off power, 2-4
 - turning on, 2-2
 - turning on power, 2-3
- robot axes only
 - mirror image shift, 10-6
 - shift, 10-19
- robot I/O, instructions, 6-101
- robot input, 3-49
- robot output, 3-49
- robot service request (RSR), 3-131, 7-26
 - setting up, 3-133
 - setup items, 3-132
- robot speed
 - AUTO mode, 1-24
 - T1 mode, 1-23
 - T2 mode, 1-24
- robot stop variation
 - Control Reliable option, 1-28
 - European controllers, 1-27
- rotation register, shape generation setup, 10-51
- rotation speed, positional data conversion, coordinates offset function, 10-111
- rotational, mirror image, 10-4, 10-5
- rotational shift, 10-16, 10-18
- ROUT Error Codes, A-103
- RS-1/RS-4 option
 - AUTO mode, 1-24
 - MODE SELECT switch, 3-31
 - T1 mode, 1-23
 - T2 mode, 1-24
- RSR, 1-30
 - See also* robot service request (RSR)
- RSR enable/disable, instruction, 6-112
- RTCP. *See* remote TCP
- RTCP frame, 3-110
 - setup
 - direct entry method, 3-110
 - three point method, 3-110
- run program, instruction, 6-125
- running, production, 7-1

S

- S-12, 1-8
- S-420iW, 1-9
- S-450, 1-10
- S-500, 1-10
- S-6, 1-8
- S-700, 1-11
- S-800, 1-11
- S-900W, 1-12
- safe position, pounce, 5-8
- safety equipment
 - AUTO mode, 1-24
 - T1 mode, 1-23
 - T2 mode, 1-24
- safety fence, 1-30
- safety signal, status, 8-15
- safety signals, 8-15
 - Belt Broken, 8-15
 - Control Reliable option, 8-15, 8-16
 - Ext E-Stop, 8-15
 - Fence Open, 8-15
 - Hand Broken, 8-15
 - Low Air Alarm, 8-15
 - Overtravel, 8-15
 - SOP E-Stop, 8-15
 - TP Deadman, 8-15
 - TP E-Stop, 8-15
 - TP Enable, 8-15
- saving
 - See also* backing up
 - axis limits, 3-146
 - files, 9-64
 - frame, to a file, 3-129
 - frame configuration, 3-114, 3-118
 - I/O, 3-54, 3-57, 3-62
 - program files, 9-29
- saving files, 9-63
 - frame setup, 9-63
 - I/O configuration, 9-63
 - macro setup, 9-63
 - mastering information, 9-63
 - password information, 9-63
 - position register, 9-63
 - register, 9-63
 - SERVO parameters, 9-63
 - system variables, 9-63
- saving frame configuration, 3-84, 3-92, 3-99, 3-104, 3-107, 3-124, 3-127, 3-130
- schedule number setup, shape generation
 - circle schedule data, 10-55
 - hexagon schedule data, 10-58
 - rectangle schedule data, 10-60
 - slot schedule data, 10-63
- schedules, program adjust, 7-29
- SCIO Error Codes, A-104
- screen
 - alarm recovery, error recovery feature, 10-149
 - printing a, 9-38
 - setting user alarm, 3-155, 3-156
- SELECT, conditional branching instructions, 6-106
- select, instruction, 6-108
- Select menu, 9-28, 9-29
- selecting
 - jog frame, 3-128
 - program files, 9-28
 - tool frame, 3-93
- selecting a program, 5-17
- Selecting a User Frame, 3-108
- selecting a user frame, 3-114, 3-117, 3-119
- semaphore, instructions, 6-125
- semi hot start, C-9
 - controller, C-9
- SENS Error Codes, A-105
- sensor
 - I/O signal cables, Auto TCP, 12-5
 - instruction, 6-129
 - instructions, 1-34, 6-4
 - string, Auto TCP, installation, 12-3
- Sensor Interface
 - programming, 11-6
 - sensor setup and hardware connections, 11-7
- Sensor Setup, Auto TCP, 12-7
- serial floppy disk, 9-17
- serial printer, 9-17
- Servo Disconnect, safety signal, Control Reliable option, 8-16
- servo parameter file, 9-53
- Servomotor, Definition of, 1-3
- servomotor, duty diagnosis, 8-38

- SERVO parameters, 9-63
- setting, system variables, 8-13
- setting up, shape generation, 10-51
- setting up a port, 9-7
- setting up interconnect I/O, 3-29
- setting user alarm screen, 3-155, 3-156
- setup
 - alarm code monitoring, 10-154
 - application, C-4
 - axis limits, 3-145
 - brake on hold, 3-151
 - brake timers, 3-148
 - CRT/KB, B-1, B-2
 - current language, 3-152
 - default device, 9-18
 - error recovery, 10-152, 10-153
 - alarm code monitoring, 10-154
 - approval DI index number item, 10-152
 - auto start max count item, 10-152
 - auto start max count R[] item, 10-152
 - automatic start feature item, 10-152
 - digital input alarms, 10-156, 10-157
 - dry run exit/entry item, 10-152
 - error recovery function item, 10-152
 - fast exit/entry feature item, 10-152
 - incomplete DO index number item, 10-152
 - items, 10-152
 - MAINT DO index number item, 10-152
 - maintenance program item, 10-152
 - reset DI index number item, 10-152
 - screen, 10-152
 - status DO index number item, 10-152
- frames, 3-1, 3-78
- I/O, 3-3, 3-43
- interconnect I/O, 3-29
- jog frame, 3-120
 - direct entry method, 3-120, 3-125
 - three point method, 3-120, 3-121
- macro command, 3-136, 3-139
- operator panel signal
 - axis limits, 3-2
 - brake on hold, 3-2
 - brake timers, 3-2
 - UOP, 3-1
- override select, 3-158, 3-160
- payload, 3-176
- port initialization, 9-3
 - location of standard and optional ports, 9-4
- production, 3-1
- Program ToolBox, 10-3, 10-184
- RTCP frame
 - direct entry method, 3-110
 - three point method, 3-110
- tool frame, 3-80
 - direct entry method, 3-81, 3-90
 - six point method, 3-81, 3-85
 - three point method, 3-81, 3-82
- user alarm, 3-155, 3-156
 - severity, 3-155, 3-157
- user frame, 3-94
 - direct entry method, 3-95, 3-105
 - four point method, 3-95
 - three point method, 3-95, 3-111
- user keys, 3-136
- severity
 - See also* error code
 - error code descriptions, A-7
- SFSPD, UOP input signals, 3-35
- shape adjust utility, shape generation, 10-71
- shape frames, shape generation, 10-75
- shape generation, 10-50
 - center position register setup, 10-51
 - circles, 10-50, 10-53
 - cut direction override setup, 10-51
 - cut direction setup, 10-51
 - cut shape macros, 10-67
 - hexagons, 10-50, 10-53
 - installation, 10-50
 - kerf width override setup, 10-51
 - kerf width setup, 10-51
 - process, 10-50
 - process macros, 10-68
 - process off macro, 10-68
 - process on macro, 10-68
 - production mode, 10-70
 - programming, 10-65
 - rectangles, 10-50, 10-53
 - rotation register setup, 10-51
 - setup, 10-51
 - shape adjust utility, 10-71
 - shape frames, 10-75
 - shape macros, 10-67
 - shape schedule register setup, 10-51
 - slots, 10-50, 10-53
 - speed override setup, 10-51
 - step through center setup, 10-51
 - teach mode, 10-70
 - using override resistors, 10-69
 - using the schedule and detail screen, 10-53
- shape information setup, procedure, 10-52
- shape macros, shape generation, 10-67

- shape schedule register, shape generation setup, 10-51
- shape setup, items, 10-51
- shift, 10-16
 - ext axes only, 10-20
 - extended axes, 10-19
 - mirror image, 10-1
 - parallel, 10-16, 10-17, 10-18
 - program base, 10-1
 - replace ext axes, 10-21
 - rotational, 10-16, 10-18
 - using the, 10-21
 - with ext axes, 10-20
 - with ext integrated, 10-19
 - with robot axes only, 10-19
- signals
 - complementary output, 3-10, 3-50, 3-63
 - UOP, 1-30
- simulating
 - group I/O, 3-59
 - I/O, 3-6, 3-12, 3-20, 3-51
 - inputs and outputs, 3-76
- simulating I/O, 3-76
- Simultaneous EV, motion option, 6-41
- single step, testing, 7-12
- singularity stop, system variable, 1-23, 1-24, 1-25, 7-17
- six point method, tool frame, 3-81, 3-85
- size, of program, 6-6
- skip
 - instruction, 6-119
 - instructions, 1-33, 6-4
- SKIP CONDITION, instruction, 6-36
- Skip, LBL[x], motion option, 6-36
- slot, I/O
 - analog, 3-5
 - digital, 3-10, 3-51, 3-59, 3-63
 - group, 3-19
 - UOP, 3-33
- slot schedule, terminology, shape generation, 10-64
- slot schedule data, shape generation
 - blend-in angle setup, 10-64
 - blend-in distance setup, 10-64
 - comment, 10-63
 - length setup, 10-63
 - overlap angle setup, 10-64
 - schedule number setup, 10-63
 - speed setup, 10-63
- start axis setup, 10-63
 - width setup, 10-63
 - x-work angle setup, 10-64
 - y-work angle setup, 10-64
- slots, shape generation, 10-50, 10-53
- soft float, error recovery limitations, 10-150
- SOP, 1-20
 - I/O, 3-3, 3-49, 8-36
- space check function, 10-26
- \$speed, \$TCPPSPEED structure, 10-181
- speed, 6-11, 6-23
 - circular motion, 6-24
 - dry run, error recovery, 10-149
 - jog, 2-6
 - joint motion, 6-23
 - linear motion, 6-24
 - maximum, program instructions, 6-118
 - programmed, 6-23
 - rotation, coordinates offset function, 10-111
 - units of, 1-31
- speed override, shape generation setup, 10-51
- speed setup, shape generation
 - circle schedule data, 10-55
 - hexagon schedule data, 10-58
 - rectangle schedule data, 10-61
 - slot schedule data, 10-63
- \$speed, \$TCPPSPEED, 10-181
- SpotTool Software, program, 1-33
- SRVO Error Codes, A-106
- standard operator panel. *See* SOP
- standard operator panel cycle start, 7-22, 7-23
- START, UOP input signals, 3-35
- start
 - automatic, error recovery feature, 10-149
 - cold, C-12
 - controlled, C-12
 - ctrl, C-12
 - init, C-12
 - performing a cold, C-7
 - performing a controlled, C-4
 - performing a controlled 2, C-6
 - performing a reinit, C-10
- start axis setup, shape generation
 - circle schedule data, 10-56
 - hexagon schedule data, 10-58
 - rectangle schedule data, 10-61
 - slot schedule data, 10-63

- START CTRL, C-3
 - START CTRL2, C-6
 - start methods
 - controlled start, C-3
 - init start, C-2
 - re-init start, C-10
 - semi hot start, C-9
 - START CTRL2, C-6
 - starting point
 - configuring I/O, 3-50
 - I/O
 - digital, 3-10, 3-51, 3-59, 3-63
 - group, 3-19
 - UOP, 3-33
 - status
 - display overview, 8-1
 - duty diagnosis, 8-38, 8-39
 - execution history, 8-34
 - indicator overview, 8-1
 - indicators, 8-2
 - memory, 8-26
 - pallet register, 8-11
 - position, 8-28
 - position register, 8-8
 - program timer, 8-18
 - register, 8-6
 - safety signal, 8-15
 - system variable, 8-13
 - user display, 8-5
 - version identification, 8-23
 - status DO, error recovery, 10-149
 - status DO index number, setup item, error recovery, 10-152
 - status indicators, 1-17
 - step path node test cycle condition, 7-10
 - step statement type test cycle condition, 7-10
 - step through center, shape generation setup, 10-51
 - stop, General Stop, Control Reliable option, 8-16
 - storage device
 - CMOS RAM, 9-2
 - disk, 9-2
 - personal computer, 9-3
 - string sensor, Auto TCP, installation, 12-3
 - sub type, 6-7
 - cond, 6-7
 - macro, 6-7
 - sub-groups, motion, 2-10
 - subprogram, branching instructions, 6-105
 - subprogram call, instruction, 6-106
 - SVON Input, safety signal, Control Reliable option, 8-16
 - SYSMACRO.SV, 9-53
 - SYSMAST.SV, 9-53
 - SYSPASS.SV, 9-53
 - SYSRDY, UOP output signals, 3-39
 - SYSSERVO.SV, 9-53
 - SYST Error Codes, A-117, A-118
 - system file, backing up, 9-55
 - system files
 - SYSMACRO.SV, 9-53
 - SYSMAST.SV, 9-53
 - SYSPASS.SV, 9-53
 - SYSSERVO.SV, 9-53
 - SYSVARS.SV, 9-53
 - system password file, 9-53
 - system variable
 - singularity stop, 1-23, 1-24, 1-25, 7-17
 - status, 8-13
 - system variable file, 9-53
 - system variables
 - displaying, 8-13
 - setting, 8-13
 - SYSVARS.SV, 9-53
- ## T
- T1, MODE SELECT switch, 7-16
 - T1 mode
 - errors, 1-23
 - locking, 1-23
 - MODE SELECT switch, 1-23, 7-12, 7-14, 7-16
 - European controllers, 1-22
 - program activation, 1-23
 - robot speed, 1-23
 - safety equipment, 1-23
 - \$t1t2_sngstp, \$PARAM_GROUP[1] structure, 1-23, 1-24, 1-25, 7-17
 - \$t1t2_sngstp, \$PARAM_GROUP[1], 1-23, 1-24, 1-25, 7-17
 - T2, MODE SELECT switch, 7-16
 - T2 mode
 - errors, 1-24
 - locking, 1-24

- MODE SELECT switch, 1-24, 7-12, 7-14, 7-16
 - European controllers, 1-22
 - program activation, 1-24
 - robot speed, 1-24
 - safety equipment, 1-24
- \$tcdelay, \$TCPPIR structure, 10-179
- \$tcdelay, \$TCPPIR, 10-179
- \$tcdelay_mon, \$TCPPSPEED structure, 10-181
- \$tcdelay_mon, \$TCPPSPEED, 10-181
- TCP, 1-31, 3-80
 - attachment device, Auto TCP, 12-3
 - fixed, positional data conversion for coordinates
 - offset function, 10-112, 10-114
 - location of, 6-20
 - orientation setup, for Auto TCP, 12-1
- TCP attachment device, Auto TCP, installation, 12-3
- TCP Orientation Setup, Auto TCP, 12-11
- TCP Points, generating automatically, 12-14
- TCP Speed Prediction, 10-176
- TCP X, Y, Z Values, Auto TCP, 12-11
- TCPP Error Codes, A-122
- \$TCPP_CFG
 - .\$group_num, 10-183
 - .\$soft_tim_enb, 10-183
 - .\$warning_enb, 10-183
- \$TCPP_CFG.\$group_num, 10-183
- \$TCPP_CFG.\$soft_tim_enb, 10-183
- \$TCPP_CFG.\$warning_enb, 10-183
- \$TCPPIR
 - .\$enable_tcp, 10-179
 - .\$tcdelay, 10-179
- \$TCPPIR.\$enable_tcp, 10-179
- \$TCPPIR.\$tcdelay, 10-179
- \$TCPPSPEED
 - .\$accel, 10-180
 - .\$motype, 10-180
 - .\$prog_speed, 10-180
 - .\$speed, 10-181
 - .\$tcdelay_mon, 10-181
 - .\$timestamp, 10-181
 - .\$vspeed, 10-182
- \$TCPPSPEED.\$accel, 10-180
- \$TCPPSPEED.\$motype, 10-180
- \$TCPPSPEED.\$prog_speed, 10-180
- \$TCPPSPEED.\$speed, 10-181
- \$TCPPSPEED.\$tcdelay_mon, 10-181
- \$TCPPSPEED.\$timestamp, 10-181
- \$TCPPSPEED.\$vspeed, 10-182
- teach mode, shape generation, 10-70
- teach pendant, 1-17
 - executing macro commands from, 3-141
 - indicators, 8-2
 - keys, 1-19
 - picture of standard, 1-18
 - screen, 1-19
 - user keys, 3-136
- teach pendant program, application files, backing up, 9-53
- teach pendant screen, printing a, 9-38
- temperature, duty diagnosis, 8-39
- terminal
 - DEC VT-220, B-1
 - emulation, B-1
 - FANUC industrialized, B-1
- termination type, 1-31, 6-11, 6-32
 - continuous, 6-33
 - fine, 6-32
- terminology, slot schedule, shape generation, 10-64
- test cycle, 7-9
 - setting up, 7-10, 7-11
 - step path node condition, 7-10
 - step statement type condition, 7-10
- test mode, error recovery feature, 10-149
- Test Mode 1. *See* T1 mode
- Test Mode 2. *See* T2 mode
- testing, 7-1
 - a program, 7-9
 - backward, 7-12
 - continuous, 7-16
 - continuously, 7-9
 - error recovery, 10-161
 - forward, 7-12
 - single step, 7-9, 7-12
 - using cycle start, 7-18
 - using the teach pendant, 7-16
- three point method
 - jog frame, 3-120, 3-121
 - RTCP frame, 3-110
 - tool frame, 3-81, 3-82
 - user frame, 3-95, 3-111
- time before/after motion option instruction, 12-1

- timer
 - instruction, 6-113
 - program, status, 8-18
 - \$timestamp, \$TCPPSPEED structure, 10-181
 - \$timestamp, \$TCPPSPEED, 10-181
 - timing diagram
 - normal operation, error recovery, 10-167
 - normal operation auto start mode, error recovery, 10-164
 - normal operation when alarm occurs, error recovery, 10-168
 - normal operation without resume program execution, error recovery, 10-165
 - resume program aborted, error recovery, 10-166
 - TOGGLE REMOTE TCP, FCTN menu item, 2-13
 - tool, coordinate system, 2-8
 - tool center point. *See* TCP
 - tool center position. *See* TCP
 - tool frame, 3-78, 3-79
 - selection of, 3-93
 - setup, 3-80
 - direct entry method, 3-81, 3-90
 - six point method, 3-81, 3-85
 - three point method, 3-81, 3-82
 - tool offset, coordinates offset function, 10-110, 10-114, 10-115
 - convert position data, 10-117
 - convert type, 10-114
 - coordinate system number setting screen, 10-114
 - end line, 10-114
 - insert line, 10-114
 - new program, 10-114
 - new UTOOL number, 10-114
 - old UTOOL number, 10-114
 - original program, 10-114
 - positional data conversion, 10-112, 10-113, 10-114
 - program name setting screen, 10-114
 - range, 10-114
 - robot fixed, 10-113, 10-114
 - start line, 10-114
 - TCP fixed, 10-112, 10-114
 - Tool_offset, motion option, 6-38, 6-124
 - Tool_offset, PR[], motion option, 6-39
 - Torch Guard, limitations, 10-141
 - touching up a program, 5-17, 5-22
 - touchup, function key, 6-14
 - TPENBL, UOP output signals, 3-39
 - TPIF Error Codes, A-124
 - turn number, display, 8-30
 - turning off the robot, 2-4
 - turning on the robot, 2-1, 2-2, 2-3
 - semi hot start, C-9
- ## U
- uframe
 - instructions, 6-123
 - offset, coordinates offset function, 10-110, 10-113, 10-117, 10-118
 - uframe offset, coordinates offset function
 - coordinate system number setting screen, 10-117
 - end line, 10-117
 - insert line, 10-117
 - new program, 10-117
 - new UTOOL number, 10-117
 - old UTOOL number, 10-117
 - original program, 10-117
 - program name setting screen, 10-117
 - range, 10-117
 - start line, 10-117
 - uframe_num, instructions, 6-122
 - unconditional branching, instructions, 6-105
 - jump, 6-105
 - subprogram, 6-105
 - undoing operations, in programs, 5-20, 5-28
 - UNLOCK PREG, position register look-ahead function instruction, 6-134
 - UOP, 1-26
 - I/O, 3-3, 3-33, 3-49
 - comments, 3-35
 - rack, 3-33
 - slot, 3-33
 - starting point, 3-33
 - inputs, 3-35
 - modular I/O board layout, 3-34
 - outputs, 3-39
 - process I/O board layout, 3-34
 - signals, 1-30
 - UOP I/O, configuring, 3-58
 - UOP production start, 7-27, 10-33
 - upper limits, axes, 3-146
 - user
 - I/O, 3-3, 3-49
 - position status, 8-28
 - user alarm
 - instruction, 6-112

setup, 3-155, 3-156, 3-157
 severity, 3-155, 3-157
 user alarms, error recovery, 10-156
 user condition param enabled, manual function detail screen, error recovery, 10-162
 User Display, displaying user screen, 8-5
 user display, status, 8-5
 user frame, 3-78
 clearing the current, 3-114
 offset, coordinates offset function. *See* UFRAME offset
 remote TCP, 2-9, 2-13
 selecting, 3-114, 3-117, 3-119
 setup, 3-94
 direct entry method, 3-95, 3-105
 four point method, 3-95
 three point method, 3-95, 3-111
 user I/O, distributed I/O, 3-49
 user keys, setting up, 3-136
 user menu, 8-5
 user operation panel start, 7-24
 user operator panel. *See* UOP
 user operator panel start, 7-24
 using DIAG, C-17
 using EMON, C-15
 using override registers, shape generation, 10-69
 utilities, using bootrom, C-14
 utility
 mirror image, 10-4
 reference position, 10-43
 shift, 10-16
 UTOOL, adjust, Program ToolBox, 10-184, 10-189
 utool, instructions, 6-123
 UTOOL adjust, Program ToolBox, 10-184, 10-189
 utool_num, instructions, 6-122

V

V-210 vision system, 6-129
 VARS Error Codes, A-131
 version identification, status, 8-23
 motor identification, 8-23
 motor information, 8-23
 servo parameters, 8-23
 software, 8-23
 via position, 6-14
 circular motion type, 6-15
 \$vspeed, \$TCPPSPEED structure, 10-182
 \$vspeed, \$TCPPSPEED, 10-182
 VT-200, B-1

W

wait, instructions, 1-33, 6-4, 6-109
 wait condition, 6-109
 wait time, 6-109
 wait condition, instructions, 6-109
 forever, 6-109
 timeout, LBL[1], 6-109
 wait release, 7-21
 wait semaphore, instructions, 6-125
 wait time, instructions, 6-109
 \$warning_enb, \$TCPP_CFG structure, 10-183
 \$warning_enb, \$TCPP_CFG, 10-183
 WHEN, condition monitor instructions, 6-135
 width setup, shape generation
 rectangle schedule data, 10-60
 slot schedule data, 10-63
 with ext axes
 mirror image shift, 10-7
 shift, 10-20
 WNDW Error Codes, A-134
 WORLD, coordinate system, 2-8
 world
 frame, 3-78
 position status, 8-28

world frame, 3–78
wrist jog, 2–9
wrist jogging, display, 2–9
wrist joint, motion option, 6–34
wrist orientation, minimizing changes in, 5–3
write protection, 6–8
writing, a program, 5–1
writing a program, 5–10

X

x–work angle setup, shape generation
circle schedule data, 10–56

hexagon schedule data, 10–58
rectangle schedule data, 10–61
slot schedule data, 10–64

XYZ, coordinate system, 2–8

Y

y–work angle setup, shape generation
circle schedule data, 10–56
hexagon schedule data, 10–58
rectangle schedule data, 10–61
slot schedule data, 10–64