



FANUC
ACADEMY

HandlingTool Operations and Programming



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

This document is based on information available at the time of its publication. While efforts have been made to be accurate, the information contained herein does not purport to cover all details or variations in hardware or software, nor provide for every possible contingency in connection with installation, operation, or maintenance. Features may be described herein which are not present in all hardware and software systems. FANUC America assumes no obligation of notice to holders of this document with respect to changes subsequently made.

FANUC America makes no representation or warranty, expressed, implied, or statutory with respect to, and assumes no responsibility for the accuracy, completeness, sufficiency, or usefulness of the information contained herein. No warranties of merchantability or fitness for purpose shall apply.

No part of this manual may be reproduced in any form. The information illustrated or contained herein is not to be reproduced, copied, downloaded, translated into another language, published in any physical or electronic format, including internet, or transmitted in whole or in part in any way without the prior written consent of FANUC America Corporation.

**Handling Tool Operations and Programming
MATAGHAND0318CE REV. F
© 2020 FANUC America Corporation
All Rights Reserved**

FANUC conducts courses on its products on a regularly scheduled basis at several locations in North America as well as on-site. For additional information contact:

FANUC America Corporation
Training Department
3900 West Hamlin Road
Rochester Hills, MI 48309

A training catalog and schedule may be downloaded from our website www.fanucamerica.com

For customer assistance, including Technical Support, Service, Parts & Part Repair, and Training Requests, contact us at 1-888-FANUC-US (1-888-326-8287).

Please send your comments and suggestions about this manual to:

training.group@fanucamerica.com

FANUC



Customer Resource Center

1-888-FANUC-US, option 1 for Robotics

www.fanucamerica.com

TECHNICAL PHONE SUPPORT * Press 1 FAX: 248-377-7832	PARTS & PART REPAIR * Press 2 FAX: 248-377-7832	ONSITE FIELD SERVICE * Press 3 FAX: 248-377-7832	PRODUCT & SYSTEMS TRAINING Press 4 FAX: 248-377-7367	MARKETING, SALES, & LITERATURE REQUESTS Press 5
<ul style="list-style-type: none"> ▪ Technical Phone Support ▪ Highly-qualified technical support specialists. ▪ Staffed 24/7 ▪ Automatic Email Notification and verification for all cases open and closed 	<ul style="list-style-type: none"> ▪ Parts/Repair Orders including quotes, order follow-up, Warranty, Repair and Down Robot. ▪ Robot software and software options 	<ul style="list-style-type: none"> ▪ Down Production Service ▪ Warranty Service ▪ Operational Audits ▪ Preventative Maintenance Services ▪ Programming Services 	<ul style="list-style-type: none"> ▪ Training class registration ▪ Consultation for non-standard training classes or on-site requests 	<ul style="list-style-type: none"> ▪ New product sales ▪ System solution sales ▪ Product and application information
<p>For best call results have:</p> <ul style="list-style-type: none"> - F# or serial number information - Hour meter reading (if available) - Software type and edition - Any error messages and LED displays (if applicable) 	<p>For best call results have:</p> <ul style="list-style-type: none"> - Part number & description (if known) - F# or serial number information - Method of payment (P.O., credit card, etc.) - Shipping & billing addresses - Repairs – problem description (any symptoms, error codes, or diagnostic LEDs that were identified) - Warranty – serial number or PO you placed original order with. 	<p>For best call results have:</p> <ul style="list-style-type: none"> - Part number & description (if known) - F# or serial number information - Method of payment (P.O., credit card, etc.) - Shipping & billing addresses - Reason for repair (any symptoms, error codes, or diagnostic LEDs that were identified) - Requested Arrival Time 	<p>For best call results have:</p> <ul style="list-style-type: none"> - Your billing address - Types of courses needed - Special requirements - F# or serial number information - Proposed schedules - Number of people attending - Student names (if available) - Method of payment (P.O., credit card, etc.) 	<p>For best call results have:</p> <ul style="list-style-type: none"> - Description of your need

***NOTES:**

A RETURN AUTHORIZATION ("RA") FROM THE SPARE PARTS DEPARTMENT IS **REQUIRED BEFORE ANY MATERIAL IS SHIPPED BACK TO FANUC ROBOTICS**. ITEMS RETURNED WITHOUT PROPER AUTHORIZATION WILL BE SUBJECT TO RETURN AT CUSTOMER EXPENSE.

ROBOT F# OR SERIAL NUMBER IS LOCATED ON THE ROBOT BASE OR OPERATIONAL PANEL. ROBODRILL SERIAL NUMBER IS LOCATED ON THE UPPER RIGHT CORNER OF RIGHT HAND DOOR OF REAR ELECTRICAL CABINET.

To register for free online services please visit: <http://robot.fanucamerica.com/Contact-us/Contact-us.aspx>



Customer Resource Center (cRc) Support Services

FREE Web Solutions

The FANUC cRc Website provides *Web Chat Support* for all our registered customers found on <https://crc.frc.com>. Additionally, the website provides free access to all product datasheets. Features include the ability to submit parts orders or request parts returns for new, warranty and certified repair. With the assistance of the parts search engine, customers can access pricing and availability information. Another key feature is order tracking with automatic email notification capabilities for all parts orders. To take advantage of our Web Solutions, register at <https://crc.frc.com> then click on the link labeled "Not Registered?"

FREE Technical Phone Support

cRc provides a multilingual, 24-hour telephone support group that possesses advanced troubleshooting skills to quickly resolve customers' technical robot issues or to simply assist with normal setup and maintenance questions. If required, we will assist with dispatching our highly-skilled service engineers to conduct onsite repairs or start-up installation support. Other routine services provided via our technical phone support include: parts identification, after-hours support, and email based support, FAQ's, error code lookup and remote diagnostics. The Technical Support Group can be reached by phone or email technicalsupport@fanucamerica.com

Training Solutions – cRc's Training Group provides 22,000+ square feet of laboratory space, current model robots, ROBODRILLS and 9 classrooms to provide you with the best "HANDS ON" training available. We also conduct classes at our regional offices or at your facility. We also offer computer-based eLearn training solutions. To register for a class or inquire about special training needs, please call or visit our website at www.fanucamerica.com/training where a complete product training schedule, course descriptions, online registration form and a wealth of other information is available.



Onsite Field Service Support – Service offerings include: local and regional dispatch, after-hours down production support, primary and resident site service engineers, installation support and programming, operation and performance audits, preventive maintenance, warranty tracking and repair, root cause analysis, and product reworks. **The Field Service Group can be reached by phone or email fieldservice@fanucamerica.com**

Parts – An inventory of over \$52 million is available to meet spare part needs. Our staff of customer service representatives are available 24/7/365 and will promptly assist you with any parts orders or inquiries. Emergency orders can be processed at any time of the day. Spare Parts can be reached by phone or emailing spareparts@fanucamerica.com, or by sending a fax to 248-377-7832.

Certified Parts Solutions – Certified repair services are available for over 1,300 different components from multiple controller platforms. Our Rochester Hills facility performs authorized repair for North America using authentic FANUC parts. After the repair is completed, it is tested to original OEM specifications. Once the repair has passed our testing, it is certified, and the warranty plus repair data are posted on-line. OEM-specified updates and upgrades are also included during the repair process at no additional cost. Contact us to obtain a return authorization or log onto the cRc Site and use our easy three-step process. For additional cost savings, core credits are available on numerous parts.

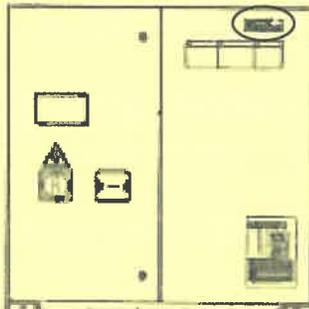
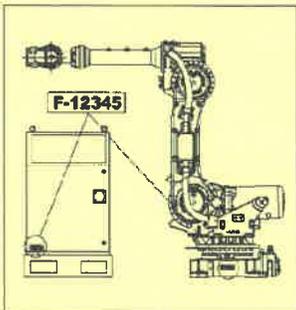
Note: All certified repairs are backed by the same one-year warranty as a new FANUC America part.

Certified Robots for Reapplication – FANUC America supports your production Robot and/or ROBODRILL for its entire production life. The following is just a list of services offered to maximize life expectancy and minimize downtime.

- Robot Evaluations
- Recertification Process
- Upgrade Options – (additional auxiliary axes, adding vision hardware/software)

*All enhanced robots are backed by a limited one-year manufacturer's warranty.

When contacting cRc Services, please have your serial number available. The F number tag locations are shown in this drawing.



FANUC America Corporation
3900 W. Hamlin Road
Rochester Hills, MI 48309-3253

Huntersville, NC
(704) 596-5121
Hoffman Estates, IL
(847) 898-6000

Lake Forest, CA
(949) 595-2700

Mason, OH
(513) 754-2400

Mississauga, Ontario,
Canada
(905) 812-2300

Montréal, Quebec,
Canada
(450) 492-9001

Aguascalientes, Mexico
52 (449) 922-8000

Sao Paulo, Brazil
55 (11) 3619-0599

ROBOGUIDE®

Robot Programming and Simulation

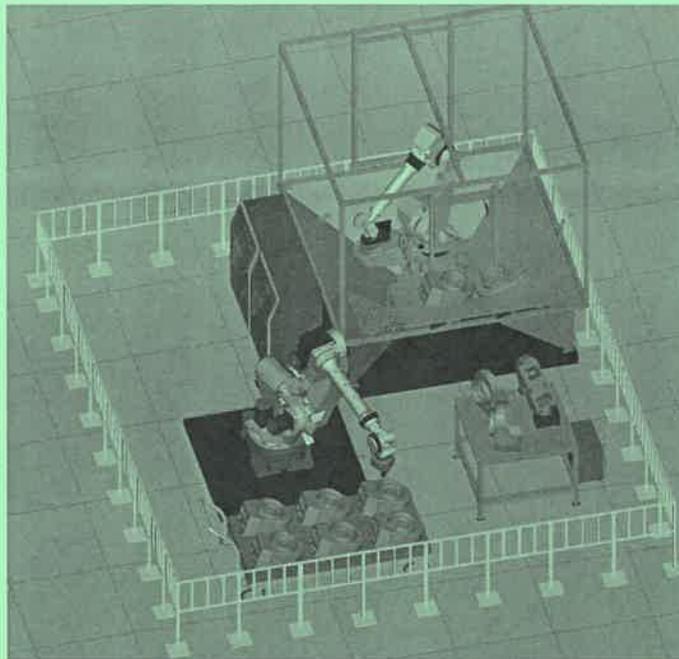
ROBOGUIDE is the leading offline programming product on the market for FANUC robots. The ROBOGUIDE family of process focused software packages allows users to create, program and simulate a robotic workcell in 3-D without the physical need and expense of a prototype workcell setup. With virtual robots and workcell models, offline programming with ROBOGUIDE reduces risk by enabling visualization of single and multi-robot workcell layouts before actual installation.

With ROBOGUIDE, sales, proposal and application engineers can import CAD models of parts, create a workcell including machines, part transfer devices and obstacles and teach robot paths to simulate the operation and performance of a multi-robot workcell. Reach verification, collision detection, accurate cycle time and robot trajectory and other system operations are provided in ROBOGUIDE's graphical virtual environment.

Virtual Robot Controller Software

True robot operation is provided using Virtual Robot Controllers which are a copy of the real software that runs on your robots. The Integrated Virtual Teach Pendant looks and operates like a real Teach Pendant. FANUC robots can be configured in many configurations with many axes. The virtual robot supports configurations including multiple arms, positioners and auxiliary axes.

The ability to create a virtual robot copy of your real robot makes it efficient to model already existing workcells. A backup of your real



robot can be used to duplicate the robot in ROBOGUIDE. This feature can be used for optimization and debug of production robots.

Cycle Time and Trajectory Profiling

Accurate cycle time reporting and trajectory plotting are essential when optimizing workcell parameters. ROBOGUIDE's cycle time profiler reports cycle time at the program, routine and motion instruction level. Robot motion trajectory tracing displays robot tool center point traces and can be plotted showing speed and accelerations along the trace.

Program Upload Download /Round Trip

Seamless data transfer between real and virtual robots. Programs can be uploaded and downloaded between real and virtual robots with no translation. Changes made in ROBOGUIDE can be directly loaded to a robot and changes on a real robot can be directly loaded to ROBOGUIDE.

Physical iPendant™ Support

Operators can learn real Teach Pendant operations quickly with a physical iPendant connection to ROBOGUIDE virtual robots. With optional interface hardware, a real iPendant can be attached and used with ROBOGUIDE. Operator training can be performed without the need for a physical robot controller and robot arm.

Workcell Modeling and Machine Building

Complete workcell simulation can be obtained. Workcells contain more than robots as moving devices. Workcells can be constructed with multiple robots, positioners, conveyors and other moving and non-moving fixtures. ROBOGUIDE supports the modeling of essentially any robot configuration including multiple arm, robot with positioner and a robot on auxiliary axes.

FANUC
Robotics

CAD to Path

Drastically reduce programming time with comprehensive CAD to Path programming. Define a feature line on a CAD model and quickly generate programs using the feature line information. Hours of manual programming can be eliminated. CAD to Path supports generation of programs for coordinated motion, remote tool center point, multi-arm coordination and other configurations.

Motion Optimization

Optimize cycle time and motion for running production robot programs to reduce cycle time. ROBOGUIDE can connect to a robot, monitor robot motion and provide direction on improving cycle time. The visual feedback of a robot's performance and the easy-to-use interface allows the robot programmer to pinpoint performance bottlenecks and take corrective action.

Process Focused Software

ROBOGUIDE builds on FANUC Robotics application process focus. Process plug-ins customize the ROBOGUIDE interface with process specific capabilities.

ROBOGUIDE - WeldPRO™

WeldPRO simulates robotic arc welding process. WeldPRO CAD to Path programming capability supports auto generation of multiple robot group coordinated motion programs with defined torch angles and process parameters. Programs and settings from the virtual workcell can be transferred to the real robot to decrease installation time.

ROBOGUIDE - HandlingPRO™

HandlingPRO is used for material handling applications including load/unload, packaging, assembly and material removal. Features of HandlingPRO include CAD to

Path programming, conveyor line tracking, machine modeling and programming.

ROBOGUIDE - PalletPRO™

PalletPRO simulation software can be used to completely build, debug and test a palletizing application offline. PalletPRO allows users to create a workcell layout, infeed and pallet stations, slip sheet and pallet dispensers. Hundreds of unit load pallet configurations can be created and visualized in 3-D using PalletPRO's built-in library of industry standard patterns. The data created in PalletPRO can be downloaded to a real robot controller containing PalletTool® software.

ROBOGUIDE - PickPRO

PickPRO lets you simulate and analyze high-speed picking processes. Visualize the operation of multi-robot systems interfaced to multiple conveyors and related equipment. It includes utilities to test the impact of various application parameters such as conveyor speed, product spacing and orientation, workload sharing among multiple robots, conveyor flow direction, robot model, equipment location, multi-tooled grippers and even exceptions such as surges and interruptions in product flow. PickPRO is an indispensable engineering and sales tool.

ROBOGUIDE - PaintPRO™

FANUC Robotics PaintPRO software is a graphical offline programming solution that simplifies robotic path, teach and paint process development. The operator automatically generates robot programs by graphically selecting the area of the part to be painted and chooses between several painting methods.

ROBOGUIDE - MotionPRO

MotionPRO is a revolutionary software which runs on a PC and when connected to a FANUC robot can provide expert assistance in optimizing robot motion. MotionPRO can potentially reduce 5-20% of cycle time and a significant amount of teaching and touch-up time.

Note: ROBOGUIDE® and PalletTool® are registered trademarks of FANUC LTD.

FANUC
Robotics

Intelligent Robot Solutions

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

Charlotte, NC
(704) 596-5121

Toronto, Canada
(905) 812-2300

Chicago, IL
(847) 898-6000

Montréal, Canada
(450) 492-9001

For sales or technical information, call:
1-800-iQ-ROBOT

Cincinnati, OH
(513) 754-2400

Aguascalientes, Mexico
52 (449) 922-8000

Los Angeles, CA
(949) 595-2700

Sao Paulo, Brazil
(55) (11) 3619-0599

marketing@fanucrobotics.com
fanucrobotics.com

Toledo, OH
(419) 866-0788

Table of Contents

1	COURSE OVERVIEW	1
1.1	COURSE GOAL	1
1.2	PREREQUISITES	1
1.3	COURSE SCHEDULE	1
1.4	OBJECTIVES	2
2	SAFETY	7
2.1	CONSIDERING SAFETY FOR YOUR ROBOT INSTALLATION	7
2.1.1	Keeping People Safe	7
2.1.2	Using Safety Enhancing Devices	8
2.1.3	Setting Up a Safe Workcell	8
2.1.4	Staying Safe While Teaching or Manually Operating the Robot	9
2.1.5	Staying Safe during Automatic Operation	10
2.1.6	Staying Safe During Inspection	11
2.1.7	Staying Safe During Maintenance	11
2.2	KEEPING MACHINE TOOLS AND EXTERNAL DEVICES SAFE	13
2.2.1	Programming Safety Precautions	13
2.2.2	Mechanical Safety Precautions	13
2.3	KEEPING THE ROBOT SAFE	13
2.3.1	Operating Safety Precautions	13
2.3.2	Programming Safety Precautions	14
2.4	WARNINGS, CAUTIONS, AND NOTES USED IN THIS MANUAL	14
2.4.1	Meaning of Warnings, Cautions, and Notes	14
2.5	DUAL CHECK SAFETY (DCS)	15
2.6	TRAINING LAB SAFETY	15
2.6.1	Lab Safety Rules	15
3	PRE-TEST	17
4	ROBOT SYSTEM	23
4.1	LEARNING OBJECTIVES	23
4.2	ROBOT SYSTEM OVERVIEW	23
4.3	MECHANICAL UNIT	24

Table of Contents

	HandlingTool
4.3.1 Motors.....	24
4.3.2 Brakes	25
4.3.3 Serial Pulse Coder.....	26
4.4 CONTROLLER	26
4.4.1 Motion Groups	28
4.4.2 Software	29
4.5 STANDARD OPERATOR PANEL.....	29
4.5.1 Mode Select Switch	30
4.6 STARTING UP THE CONTROLLER	32
4.6.1 Conducting a Cold Start.....	32
4.6.2 Cycle Power from a Powered Up Controller.....	33
4.6.3 Other Start Up Methods	34
4.7 POWERING DOWN THE CONTROLLER	35
4.8 CHAPTER REVIEW	36
5 IPENDANT	39
5.1 LEARNING OBJECTIVES	39
5.2 IPENDANT OVERVIEW	39
5.3 IPENDANT SWITCHES AND KEYS	40
5.3.1 ON / OFF Switch (ENABLE / DISABLE Switch)	40
5.3.2 Deadman Switch.....	40
5.3.3 Emergency Stop Button	41
5.3.4 iPendant Keys.....	42
5.4 STATUS BAR DISPLAYS	45
5.4.1 LED Indicators	45
5.5 SCREEN NAVIGATION	46
5.5.1 Touch Screen Feature	46
5.5.2 Display Shortcuts.....	46
5.5.3 Split Screen	47
5.5.4 Menu Navigation.....	50
5.5.5 Function Menu	51
5.5.6 i Key	52
5.6 IPENDANT FUNCTIONS	52
5.6.1 Menu Favorites	53

Table of Contents

Handling Tool

5.6.2	Software Keyboard	54
5.6.3	Quick / Full Menus	54
5.7	CHAPTER REVIEW	56
6	JOGGING.....	59
6.1	LEARNING OBJECTIVES	59
6.2	JOGGING OVERVIEW	59
6.2.1	Jog Speed	59
6.2.2	Coordinate System	60
6.2.3	Jogging in Joint.....	61
6.2.4	Right Hand Rule and Cartesian Coordinate Systems.....	62
6.2.5	Jogging in World Frame.....	62
6.3	JOGGING THE ROBOT.....	63
6.4	POSITIONAL DATA.....	64
6.5	CHAPTER REVIEW	66
7	FILE MANIPULATION.....	69
7.1	LEARNING OBJECTIVES	69
7.2	CONTROLLER MEMORY OVERVIEW	69
7.3	TYPES OF BACKUPS.....	70
7.4	PREPARING TO CREATE A BACKUP	71
7.4.1	Setting the Default Device.....	71
7.4.2	Formatting the Device.....	72
7.4.3	Subdirectories.....	74
7.5	FILE BACKUP.....	75
7.6	FILE RESTORE.....	78
7.7	IMAGE BACKUP.....	81
7.8	IMAGE RESTORE.....	83
7.9	GENERATING A DIRECTORY	85
7.10	CHAPTER REVIEW	87
8	INITIAL SETUP	91
8.1	LEARNING OBJECTIVES	91
8.2	AXIS LIMITS SETUP	91
8.3	AXIS LIMITS SOFTWARE SETTINGS.....	92

Table of Contents

HandlingTool

8.3.1	Axis Limits Software Settings Faults	93
8.4	PAYLOAD	93
8.4.1	Payload Setup	94
8.4.2	Payload Instructions.....	95
8.4.3	Arm Load Information	96
8.5	MASTERING OVERVIEW.....	97
8.5.1	Verify Mastering.....	97
8.6	CHAPTER REVIEW	99
9	ALARMS	103
9.1	LEARNING OBJECTIVES	103
9.2	FAULT AND MESSAGE DIAGNOSIS	103
9.2.1	Displaying the Alarm Log	103
9.2.2	Displaying the Alarm History	105
9.2.3	Diagnosing a Fault.....	105
9.3	DUAL CHECK SAFETY (DCS) OPTION	106
9.4	SINGULARITY.....	107
9.5	CHAIN FAILURE DETECTION ERROR.....	107
9.6	VERSION IDENTIFICATION	108
9.6.1	Robot F-Number	108
9.6.2	Version Identification Status.....	108
9.7	CHAPTER REVIEW	110
10	FRAMES	113
10.1	LEARNING OBJECTIVES.....	113
10.2	FRAMES OVERVIEW	113
10.2.1	How are Frames Used	114
10.3	FRAME TYPES	114
10.4	WORLD FRAME.....	115
10.5	TOOL FRAMES.....	115
10.6	CREATING A TOOL FRAME.....	117
10.6.1	Three Point Method	118
10.6.2	Four Point Method	121
10.6.3	Six Point (XY / XZ) Method	127
10.6.4	Direct Entry Method	131

Table of Contents

HandlingTool

10.7	USER FRAMES.....	135
10.8	CREATING A USER FRAME.....	136
10.8.1	Three Point Method	136
10.8.2	Four Point Method	140
10.8.3	Direct Entry Method	144
10.9	JOG FRAMES.....	147
10.10	CREATING JOG FRAMES	147
10.10.1	Three Point Method.....	148
10.10.2	Direct Entry Method.....	151
10.11	SAVING AND RESTORING FRAME DATA.....	153
10.11.1	Saving Frame Data	154
10.11.2	Restoring Frame Data	155
10.12	CHAPTER REVIEW	157
11	CREATING A PROGRAM.....	161
11.1	CHAPTER OBJECTIVES.....	161
11.2	MOTION PROGRAM OVERVIEW	161
11.3	SELECT SCREEN OVERVIEW.....	161
11.3.1	Select Screen Soft Keys	161
11.3.2	Selecting a Program	162
11.4	EDIT SCREEN OVERVIEW	162
11.4.1	Edit Screen Soft Keys.....	163
11.5	CREATING A PROGRAM	163
11.6	PROGRAM DETAIL INFORMATION	165
11.6.1	Creation Date, Modification Date, Copy Source, Positions, and Program Size...166	
11.6.2	Program Name	166
11.6.3	Sub Type	167
11.6.4	Program Comment.....	168
11.6.5	Group Mask	168
11.6.6	Write Protection	168
11.6.7	Ignore Pause	169
11.6.8	Stack Size.....	170
11.7	ADDING MOTION INSTRUCTIONS TO A PROGRAM.....	170
11.7.1	Motion Instructions and Frames.....	171

Table of Contents

	HandlingTool
11.7.2 Default Motion Instructions.....	173
11.8 TESTING A PROGRAM.....	174
11.8.1 Single STEP Testing a Program	175
11.8.2 Continuous Testing.....	176
11.8.3 Aborting a Program.....	176
11.8.4 Running a Program in AUTO	177
11.9 TOUCHING UP A POINT.....	178
11.9.1 Set New ID	179
11.9.2 Enter UTool / UFrame.....	180
11.10 SAVING AND RESTORING PROGRAM FILES	180
11.10.1 Saving Program Files	180
11.10.2 Restoring Program Files.....	182
11.11 CHAPTER REVIEW.....	183
12 MOTION INSTRUCTIONS.....	189
12.1 CHAPTER OBJECTIVES.....	189
12.2 MOTION INSTRUCTION OVERVIEW.....	189
12.3 MOTION TYPES.....	190
12.3.1 Joint Motion	190
12.3.2 Linear Motion.....	190
12.3.3 Changing Motion Types	190
12.3.4 Circular Motion.....	191
12.4 POSITION INFORMATION.....	193
12.4.1 Configuration	193
12.5 POSITION TYPES	194
12.6 POSITION	194
12.7 POSITION REGISTERS	194
12.7.1 Example of Position Registers	195
12.7.2 Creating Position Registers.....	196
12.7.3 Viewing Position Register Data.....	196
12.7.4 MOVE TO a Position Register	197
12.7.5 Adding a Position Register to a Program	197
12.7.6 Position Register Representation.....	198
12.7.7 Creating a Truly Global Position Register	199

Table of Contents

HandlingTool

12.8	SPEED	200
12.8.1	Motion Statement Speed Values.....	200
12.8.2	General Override	200
12.9	TERMINATION TYPE	201
12.9.1	Fine Termination.....	201
12.9.2	Continuous Termination.....	201
12.10	CHAPTER REVIEW	203
13	EDIT COMMAND	209
13.1	CHAPTER OBJECTIVES.....	209
13.2	EDIT COMMAND MENU	209
13.2.1	Accessing the Edit Command Menu	210
13.3	INSERTING BLANK LINES	210
13.4	DELETING LINES	212
13.5	COPY AND PASTE	213
13.5.1	Pasting Options	213
13.5.2	Reverse Motion Pasting.....	214
13.5.3	Execute a Copy and Paste Operation	216
13.6	REPLACING INSTRUCTIONS.....	217
13.7	RENUMBERING POSITIONS	218
13.8	DISPLAYING COMMENTS	220
13.9	UNDOING OPERATIONS	220
13.9.1	Undo Limitations.....	221
13.10	REMARKING INSTRUCTIONS	222
13.11	DISPLAYING INSTRUCTIONS IN COLOR	223
13.12	CHAPTER REVIEW	224
14	INPUTS AND OUTPUTS (I/O)	227
14.1	CHAPTER OBJECTIVES.....	227
14.2	I/O OVERVIEW.....	227
14.3	TYPES OF I/O	227
14.3.1	Digital	228
14.3.2	Group	228
14.3.3	Analog	228
14.3.4	User Operator Panel I/O	228

Table of Contents

	HandlingTool
14.3.5 Standard Operator Panel (SOP) I/O.....	228
14.3.6 Robot I/O	228
14.4 I/O HARDWARE.....	229
14.4.1 Model A I/O.....	229
14.4.2 Model B I/O.....	229
14.4.3 Other Types.....	229
14.5 SETTING UP I/O (CONFIGURING).....	230
14.5.1 Rack	230
14.5.2 Slot	230
14.5.3 Starting Point (Digital) / Channel (Analog).....	231
14.5.4 Configuring I/O.....	231
14.5.5 Digital I/O Comments, Complimenting, and Polarity.....	232
14.6 GROUP I/O.....	234
14.6.1 Configuring Group I/O.....	235
14.7 CONTROLLING I/O.....	237
14.7.1 Monitoring I/O	238
14.7.2 Forcing I/O.....	238
14.7.3 Simulating I/O	240
14.8 SAVING I/O CONFIGURATION DATA	241
14.9 CHAPTER REVIEW	242
15 BRANCHING.....	247
15.1 CHAPTER OBJECTIVES.....	247
15.2 BRANCHING INSTRUCTIONS OVERVIEW	247
15.3 LABEL DEFINITION INSTRUCTION	247
15.4 UNCONDITIONAL BRANCHING	247
15.4.1 Jump Instruction	248
15.4.2 CALL Instruction	249
15.5 DATA REGISTERS	251
15.5.1 Direct Addressing Data Register Instructions.....	251
15.5.2 Performing Arithmetic Operations with Register Instructions.....	253
15.6 CONDITIONAL BRANCHING.....	257
15.6.1 IF Instructions	257
15.6.2 Types of IF Instructions.....	257

Table of Contents

HandlingTool

15.6.3	IF Instruction Limitations	258
15.6.4	Programming an IF Instruction	258
15.6.5	SELECT Instructions.....	261
15.6.6	Programming a SELECT Instruction	262
15.7	WAIT INSTRUCTIONS.....	265
15.7.1	WAIT time.....	265
15.7.2	WAIT Conditions.....	266
15.7.3	Programming a WAIT Instruction	267
15.8	CHAPTER REVIEW	269
16	MACROS.....	275
16.1	CHAPTER REVIEW	275
16.2	MACRO OVERVIEW	275
16.3	CREATING A MACRO	275
16.4	MACRO ASSIGNMENTS.....	277
16.4.1	Teach Pendant User Key Assignments.....	277
16.4.2	MANUAL FUNCTIONS Assignments.....	278
16.4.3	Input Signal Assignment	278
16.4.4	Assigning the Macro	279
16.5	MACRO COMMAND INSTRUCTION	282
16.6	CHAPTER REVIEW	284
17	PR OFFSETS AND MISC. INSTRUCTIONS	287
17.1	CHAPTER OBJECTIVES.....	287
17.2	POSITION REGISTERS OVERVIEW	287
17.3	POSITION REGISTERS INSTRUCTIONS.....	287
17.3.1	PR[x] Position Register Instructions	288
17.3.2	PR [i, j] Position Register Element Instructions	289
17.3.3	Position Register Element Instruction Limitations.....	291
17.3.4	Programming a Position Register Instruction	291
17.4	OFFSET, PR MOTION OPTION INSTRUCTION.....	294
17.4.1	Programming an OFFSET, PR [x] Instruction.....	295
17.5	MISCELLANEOUS INSTRUCTIONS.....	297
17.5.1	Timer Instruction	297
17.5.2	OVERRIDE Instruction.....	297

Table of Contents

HandlingTool

17.5.3 Remark Instruction.....	298
17.5.4 Message Instruction.....	298
17.6 CHAPTER REVIEW	299
18 POST TEST.....	307

1 COURSE OVERVIEW

This course is intended for an operator, technician, engineer or programmer who must setup, record and/or troubleshoot programs on a HandlingTool software package. The course covers the Robot Operations outline intermixed with the tasks required to set up the HandlingTool application, test, run, and refine the program and production setup.

Recommended safety procedures are integrated into all training exercises.

The course consists of lectures, chapter reviews, demonstrations and a series of lab exercises designed to reinforce what the student has learned.

In addition to lab exercises, a pre-test and a post-test are used to measure mastery of objectives.

1.1 Course goal

Upon successful completion of this course, the student will be able to perform the following:

- Power up and jog the robot.
- Execute Teach Pendant Operations.
- Set Robot Payload.
- Perform File and Image Backups.
- Save Individual Files.
- Recover from common program and robot faults.
- Create TOOL, USER, and JOG Frames.
- Execute production operations.
- Create, modify and execute a material handling program.
- Edit Programs.
- Monitor, force, and simulate input and output signals.
- Program Branching Instructions.
- Program Position Register Instructions.
- Create and execute Macros.

1.2 Prerequisites

None; although it may be easier if the student has taken the Robot Operations class first.

1.3 Course Schedule

- Session 1 (Day 1, Morning)
 - Introductions
 - Safety
 - Pre-Test
 - Robot Overview
 - iPendant

- Session 2 (Day 1, Afternoon)
 - Jogging
 - File Manipulation
 - Initial Setup
- Session 3 (Day 2, Morning)
 - Alarms
 - Frames
- Session 4 (Day 2, Afternoon)
 - Frames
 - Creating a Program
 - Edit Command
- Session 5 (Day 3, Morning)
 - Motion Instructions
 - Inputs/Outputs
- Session 6 (Day 3, Afternoon)
 - Branching
- Session 7 (Day 4, Morning)
 - Branching
 - Macros
- Session 8 (Day 4, Afternoon)
 - PR Offset and Miscellaneous Instructions
 - Final Exercise
 - Post Test
 - Course Reviews

1.4 Objectives

Students successfully completing this course will be able to:

- Robot System
 - Identify the components that make up a Robot System
 - Identify the Major and Minor Axes
 - Have a general understanding of SERVO Motors and Serial Pulse Coders
 - Identify the controller and components

- Understand motion groups and extended axes
- Understand the role of the robots software
- Identify the uses of each item on the Standard Operators Panel (SOP)
- Power on the Robot System
- iPendant
 - Receive an iPendant Overview
 - Learn the functions of the iPendant switches and keys
 - Learn how to navigate the iPendant screens
 - Perform iPendant Operations
- Jogging
 - Jog the Robot in Joint and World
 - View Positional Data
 -
- File Manipulation
 - Controller Memory Overview
 - Understand the Types of Backups
 - Perform a File Backup and Restore
 - Perform an Image Backup and Restore
- Initial Setup
 - Setup Robot Axis Limits
 - Understand Payload
 - Verify Mastering
- Alarms
 - Diagnose Faults and Messages
 - Recover from common errors
- Frames
 - Understand a 3D Cartesian Coordinate System
 - Learn the different Frame Types
 - Create Tool Frames
 - Create User Frames
 - Create Jog Frames
 - Save Frame Data

- Creating a Program
 - Understand what a motion program is
 - Select Screen Overview
 - Edit Screen Overview
 - Create a Program
 - Understand Program Detail information
 - Record positions in a program
 - Test a Program
- Motion Instructions
 - Elements that describe a motion instruction
 - Understand differences between Linear and Joint Motion
 - Position Information Overview
 - Understand the difference between a Position and a Position Register
 - Understand Robot Speed Values
 - Learn the difference between FINE and Continuous Termination
- Edit Command
 - Inserting blank lines into a program.
 - Deleting lines from a program.
 - Copying and Pasting Instructions within a program.
 - Replacing items within a program.
 - Renumbering Position ID's.
 - Displaying Comments within a program.
 - Undo commands.
 - Remarking lines in a program.
 - Displaying color backgrounds
 - Displaying I/O status.
- Inputs and Outputs
 - I/O Overview
 - Types of I/O
 - Types of I/O Hardware
 - Configuring I/O
 - Configure Group I/O

- Monitor and Force I/O
- Branching
 - Branching Instruction Overview
 - Program Unconditional Branching instructions
 - Program using Data Registers
 - Program using Conditional Branching Instructions
 - Program using Wait Instructions
- Macros
 - Create a Macro Program
 - Assign a Macro
 - Use Macro Instruction
- PR Offsets and Miscellaneous
 - Apply Position Register Instructions
 - Apply Position Register [i, j] Instructions
 - Apply Position Register Offset Motion Option Instructions
 - Utilize Miscellaneous Instructions

2 SAFETY

FANUC America Corporation 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 America Corporation 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 America Corporation systems.

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

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

FANUC America Corporation therefore, recommends that all personnel who intend to operate, program, repair, or otherwise use the robotics system be trained in an approved FANUC America Corporation 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.

2.1 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

2.1.1 *Keeping People Safe*

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

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

2.1.2 *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

2.1.3 *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 America Corporation training course(s) related to your application. Never permit untrained personnel to operate the robots.
- Install a lockout device that uses an access code to prevent unauthorized persons from operating the robot.
- Use anti-tie-down logic to prevent the operator from bypassing safety measures.
- Arrange the workcell so the operator faces the workcell and can see what is going on inside the cell.
- Clearly identify the work envelope of each robot in the system with floor markings, signs, and special barriers. The work envelope is the area defined by the maximum motion range of the robot, including any tooling attached to the wrist flange that extend this range.
- Position all controllers outside the robot work envelope.
- Never rely on software or firmware based controllers as the primary safety element unless they comply with applicable current robot safety standards.

- Mount an adequate number of EMERGENCY STOP buttons or switches within easy reach of the operator and at critical points inside and around the outside of the workcell.
- Install flashing lights and/or audible warning devices that activate whenever the robot is operating, that is, whenever power is applied to the servo drive system. Audible warning devices shall exceed the ambient noise level at the end-use application.
- Wherever possible, install safety fences to protect against unauthorized entry by personnel into the work envelope.
- Install special guarding that prevents the operator from reaching into restricted areas of the work envelope.
- Use interlocks.
- Use presence or proximity sensing devices such as light curtains, mats, and capacitance and vision systems to enhance safety.
- Periodically check the safety joints or safety clutches that can be optionally installed between the robot wrist flange and tooling. If the tooling strikes an object, these devices dislodge, remove power from the system, and help to minimize damage to the tooling and robot.
- Make sure all external devices are properly filtered, grounded, shielded, and suppressed to prevent hazardous motion due to the effects of electro-magnetic interference (EMI), radio frequency interference (RFI), and electro-static discharge (ESD).
- Make provisions for power lockout/tagout at the controller.
- Eliminate pinch points. Pinch points are areas where personnel could get trapped between a moving robot and other equipment.
- Provide enough room inside the workcell to permit personnel to teach the robot and perform maintenance safely.
- Program the robot to load and unload material safely.
- If high voltage electrostatics are present, be sure to provide appropriate interlocks, warning, and beacons.
- If materials are being applied at dangerously high pressure, provide electrical interlocks for lockout of material flow and pressure.

2.1.4 *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 iPendant 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

2: Safety

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 America Corporation 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 iPendant DEADMAN switch for proper operation, and place the robot in teach mode. Take the iPendant with you, turn it on, and be prepared to release the DEADMAN switch. Only the person with the iPendant 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.

2.1.5 *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.

HandlingTool

- 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, iPendant, or emergency stop stations around the workcell.

2.1.6 *Staying Safe During Inspection*

When inspecting the robot, be sure to

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

2.1.7 *Staying Safe During Maintenance*

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

- Never enter the work envelope while the robot or a program is in operation.
- Before entering the work envelope, visually inspect the workcell to make sure no potentially hazardous conditions exist.
- Never wear watches, rings, neckties, scarves, or loose clothing that could get caught in moving machinery.
- Consider all or any overlapping work envelopes of adjoining robots when standing in a work envelope.

2: Safety

HandlingTool

- Test the iPendant 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 iPendant with you, press the DEADMAN switch, and turn the iPendant 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 an applicator bell cup can continue to spin at a very high speed even if the robot is idle. Use protective gloves or disable bearing air and turbine air before servicing these items.
- 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 main 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.

HandlingTool

- 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.

2.2 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.

2.2.1 *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.

2.2.2 *Mechanical Safety Precautions*

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

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

2.3 KEEPING THE ROBOT SAFE

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

2.3.1 *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 iPendant.

2: Safety

- Make sure the work envelope is clean and free of oil, water, or debris.
- Use circuit breakers to guard against electrical overload.

2.3.2 Programming Safety Precautions

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

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

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

2.4 WARNINGS, CAUTIONS, AND NOTES USED IN THIS MANUAL

Throughout the student training manual you will notice a number of Warning, Caution, and Notes. These items are placed to draw the attention of the student to an item that will either keep them safe, inform of an issue that could be present, or just as a reminder. It is vital that you take notice of these items throughout the student training guide. If there are any questions or concerns relating to any warnings or cautions ensure you bring these items to the attention of your instructor.

2.4.1 Meaning of Warnings, Cautions, and Notes



WARNING

Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use. In situations where inattention could cause either personal injury or damage to equipment, a warning notice is used.



CAUTION

Caution notices are used in this publication where equipment might be damaged if care is not taken.

 **NOTE:** Note notices are used in this publication merely to call attention to information that is especially significant to understanding and operating the equipment.

2.5 Dual Check Safety (DCS)

Position/Speed Check features check the speed and position data of motors with two independent CPUs in the robot controller. These functions can detect position and speed errors immediately and shut down the motor power by two independent channels. Safety data and processes are cross-checked by two CPUs. Self-diagnosis of safety hardware and software is executed periodically to prevent potential failure accumulation.

DCS Position/Speed Check features do not need additional external sensors to monitor speed and position. Only the built-in servo motor sensors are used for this function. (To use the OPSFTY safety inputs, external electrical circuits are required.)

DCS functions are certified to meet the requirements of International Standard ISO13849-1 by a notarized body

Dual Check Safety (DCS) is a feature that consists of the following safety functions:

- Emergency Stop Control (Standard)
- Position / Speed Check (Option)
 - Joint Position Check
 - Joint Speed Check
 - Cartesian Position Check
 - Cartesian Speed Check
 - T1 Mode Speed Check



2.6 TRAINING LAB SAFETY

While working on the robots in the training or the classroom robots it is vital to heed all safety warning posted on the safety fence or near the robot, as well as all verbal instructions from your instructor. Failure to follow posted safety rules may result in your removal from the training course.

2.6.1 Lab Safety Rules

In the training lab:

- Do not work on any robot unless an instructor is present.
- Keep controller doors closed and locked at all times, unless participating in an Electrical Maintenance class.
- Lock out controllers when not in the lab.
- Keep hands clear of tooling.
- When entering the workcell pay particular attention to the following:
 - Only one person at a time may enter the workcell.
 - The robot controller must be switched to T1 (TEACH) Mode.

2: Safety

HandlingTool

- You must have an enabled (turned on) iPendant with you at all times.
- Do not place any body part between the robot and a fixed object.
- **NO ONE** is permitted in the work cell when the robot is in Auto Mode.

3 PRE-TEST

Name: _____ Date: _____

This test is designed to evaluate your knowledge of FANUC America Corporation HandlingTool Operations and Programming prior to your taking this course. Each question has four possible answers and a place to indicate, I don't know. Choose the one that best answers the question. There is only one right answer. This test tells us a little about you and helps us to tailor the course to meet your needs. We appreciate you're taking the time to complete it.

Score: ___ / 100%

- 1. Enabling the iPendant gives motion control solely to whom?**
 - A The PLC
 - B The Controller
 - C The Operator (person with the pendant)
 - D Andy from accounting
 - E I don't know

- 2. Assuming standard setup, if you jog the robot in World -Y which direction will the TCP move?**
 - A Straight out front
 - B Up
 - C To the robots right
 - D To the robots Left
 - E I don't know

- 3. What key is depressed to switch between the different jog methods?**
 - A POSN
 - B MENU
 - C FCTN
 - D COORD
 - E I don't know

3: Pre-Test

4. **What two buttons display the Jog Menu**
 - A Shift and Display
 - B Menu and any jog key
 - C Shift and COORD
 - D Group and COORD
 - E I don't know
5. **What type of memory is saved when performing an Image Backup?**
 - A F-ROM Only
 - B SRAM Only
 - C F-ROM and SRAM
 - D DRAM
 - E I don't know
6. **Singularity error can be corrected by moving?**
 - A Joint 2 (+/-) 10 degrees
 - B Joint 3 (+/-) 15 degrees
 - C Joint 5 (+/-) 10 degrees
 - D Joint 6 (+/-) 15 degrees
 - E I don't know
7. **Where is the default Tool Center Point?**
 - A Intersection of J1 and J2
 - B Center of the Faceplate
 - C Tip of the default tool
 - D FANUC robots do not use Tool Center Points
 - E I don't know
8. **What type of I/O handles communication between the controller and the robot?**
 - A Digital I/O
 - B UOP I/O
 - C Robot I/O
 - D SOP I/O
 - E I don't know

- 9. Positional data is dependent on what two frames?**
- A World and User
 - B Jog and Tool
 - C User and Tool
 - D Tool and World
 - E I don't know
- 10. Which type of instruction defines for the robot how to move, where to move, how fast to move, and how to stop?**
- A Move Instruction
 - B Logic Instruction
 - C Motion Instruction
 - D Kinematics Instruction
 - E I don't know
- 11. Which type of pasting will paste the motion instructions in order and renumber the Position ID's based on first available Position ID?**
- A LOGIC
 - B POSITION
 - C POS_ID
 - D R_POS
 - E I don't know
- 12. Which Mode limits all Cartesian speed to less than 250 mm/sec and joint speed to less than 10%?**
- A Auto Mode
 - B T2 Mode
 - C T1 Mode
 - D Neither, this is a system variable setting
 - E I don't know

3: Pre-Test

13. Which motion type moves the TCP in a straight line from the start position to the destination position?
- A Linear
 - B Joint
 - C Circular
 - D Spline
 - E I don't know
14. Which Position must the DEADMAN Switch be held to enable motion?
- A Released
 - B Centered
 - C Fully Depressed
 - D The iPendant does not have a DEADMAN Switch
 - E I don't know
15. Before forcing an input it must be _____?
- A Simulated
 - B Named
 - C Part of a Group
 - D Inputs can never be forced
 - E I don't know
16. What type of memory is saved when performing a File Backup?
- A F-ROM Only
 - B SRAM Only
 - C F-ROM and SRAM
 - D DRAM
 - E I don't know
17. The six point method of defining a Tool Frame defines the Location and _____ of a Tool Frame.
- A Distance
 - B Orientation
 - C Operation
 - D Offset
 - E I don't know

18. What are the two possible actions of an IF/SELECT statement?

- A Select & Set an output
- B Set an Output & CALL
- C JMP LBL & CALL
- D JMP LBL & WAIT
- E I don't know

19. The motion option OFFSET, PR [x] is offset relative to the orientation of what Frame?

- A TOOL
- B JOG
- C WORLD
- D USER
- E I don't know

20. Which Macro Assignments will execute a Macro in Manual Mode?

- A DI, RI, UI, SOP
- B UK, SU, MF
- C Macros can only run in AUTOMATIC Mode
- D FANUC software does not support Macros
- E I don't know

4 ROBOT SYSTEM

4.1 Learning Objectives

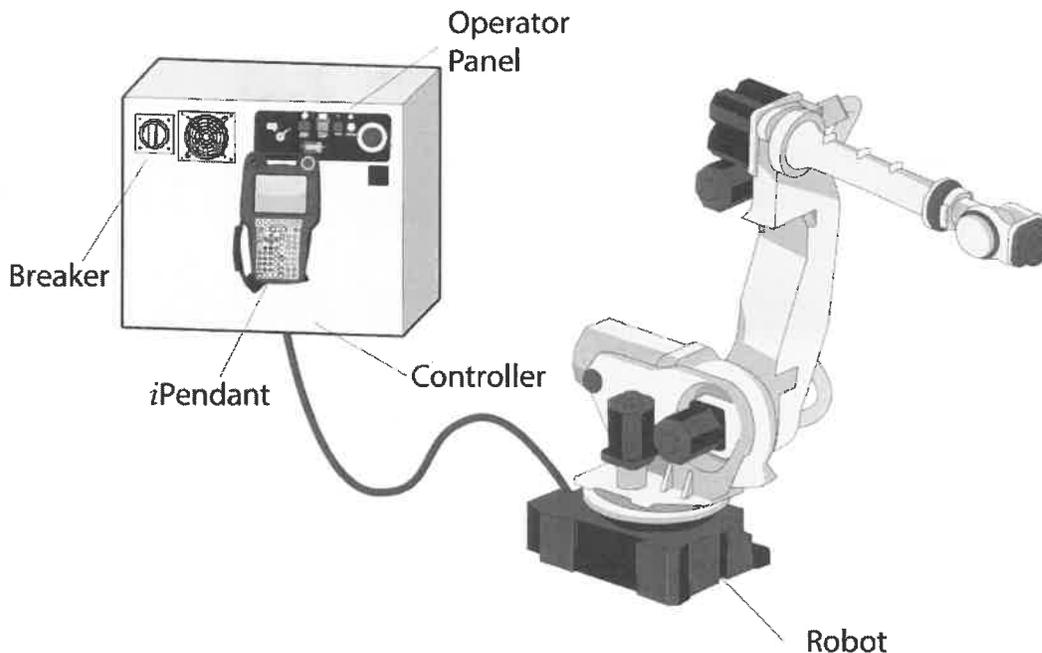
After completing this chapter you will be able to:

- Identify the components that make up a Robot System
- Identify the Major and Minor Axes
- Have a general understanding of SERVO Motors and Serial Pulse Coders
- Identify the controller and components
- Understand motion groups and extended axes
- Understand the role of the robots software
- Identify the uses of each item on the Standard Operators Panel (SOP)
- Power on the Robot System

4.2 Robot System Overview

The robot system consists of the robot (mechanical unit), the controller, and FANUC software. Due to the variation of robotic applications, FANUC makes a variety of robot mechanical units that will work with the R-30iB and R30iB Plus controller.

The image below displays an example of a robot system: the robot and controller. Your system layout might vary depending on the kind of equipment you are using. Each of the items depicted below described in more detail in this chapter.

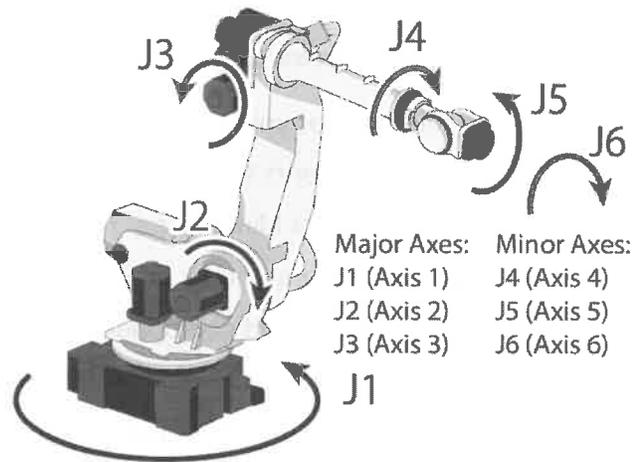


4: Robot System

4.3 Mechanical Unit

A robot is a series of mechanical links driven by servomotors. The area at each junction between the links is a joint, or axis.

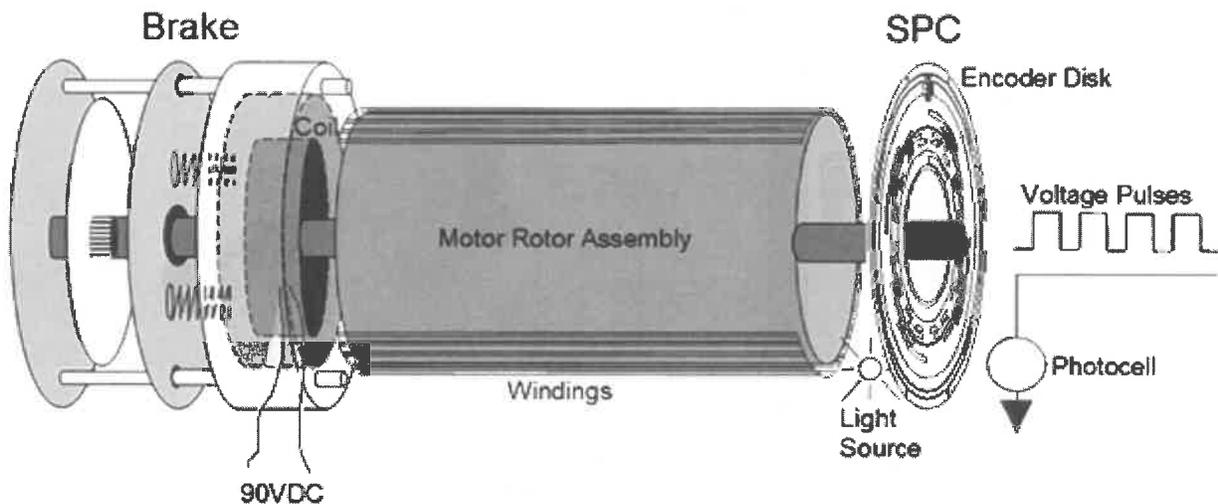
- The first three axes (1, 2, and 3) make up the major axes.
- The last three axes (4, 5, and 6) are the minor axes.
- The movements are rotational twisting, up-and-down, and side-to-side motions
- A robot is classified by the number of linear and rotational axes.



NOTE: A variety of FANUC robot models can be used. Contact your FANUC representative for more information about the kinds of robots you can use.

4.3.1 Motors

Each axis is driven by a FANUC AC servo motor. The motors are driven by a pulse width modulated (PWM) signal delivered by the servo amplifier. The motor consists of a permanent magnet rotor surrounded by windings as shown below.



4: Robot System

HandlingTool

4.3.2 Brakes

Most, although not all, robots have motors that are equipped with a brake. The brake typically consists of an armature controlled by a 90VDC brake coil. The mechanical construction of the brake is shown in below. The brake is built into the motor and cannot be replaced as a separate unit.

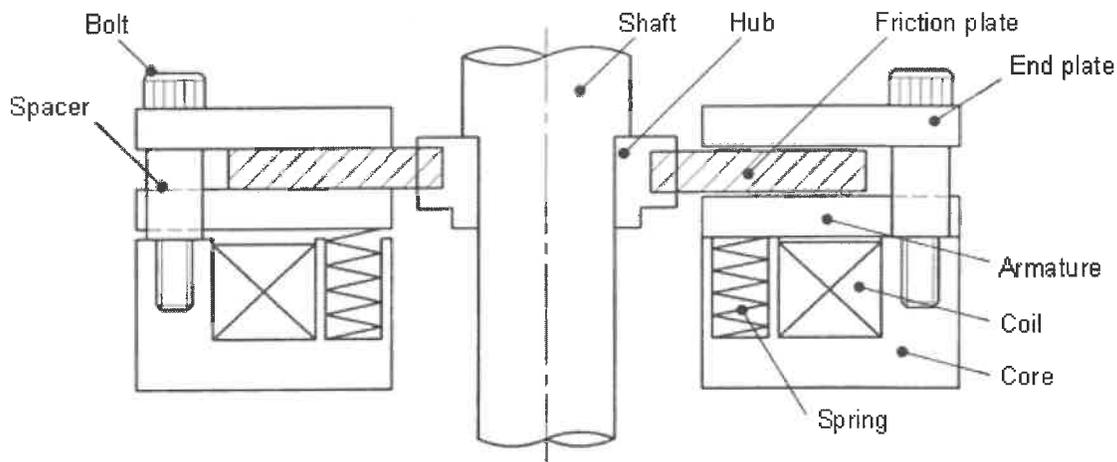
Below are a few items to note as they relate to the motors:

- Can be damaged from improper lubrication.

NOTE: Refer to the specific Mechanical Unit Operators or Maintenance Manual for lubrication guidelines.

- Brake failure is typically from lack of current at the brake as opposed to failure of the brake itself.
- Internal brake is applied mechanically and released electrically.
- Brake torque is caused by catching friction plate between armature and end plate with spring force.

(Friction plate and hub are coupled by spline; hub and shaft are bonded by shrinkage fit.)



- **Power OFF
(Applied Mechanically)**
Armature is pushed by spring force, and then friction plate is caught between armature and end plate. Friction plate is held and shaft is fixed.

- **Power ON
(Released Electrically)**
Armature is pulled toward core by electromagnet force to exceed spring force. Friction plate is released and shaft is allowed to rotate.

WARNING

DO NOT perform motor maintenance without proper training.

4: Robot System

4.3.3 Serial Pulse Coder

The serial pulse coder (SPC) is a FANUC proprietary encoder used to track position and speed for each robot motor. The SPC provides its feedback signal by means of a serial data packet. This data is sent first to the servo amplifier and then to the CPU via the axis control card. In addition to position/speed data, this data also includes status information such as the condition of the motor thermal.

In its most basic form the SPC consists of an optical disc, a light source, and a sensor to detect the light. The light passes through the disc at strategic places as the disc turns and is detected by the sensor. At any point in time the position of the axis can be determined by observing this resultant signal.

- Each motor has a SPC
- Encoder pulse counts are stored in robot memory when the robot is powered down.
- Encoder pulse count is also stored at each SPC, and maintained by batteries found at the base of the robot.



4.4 Controller

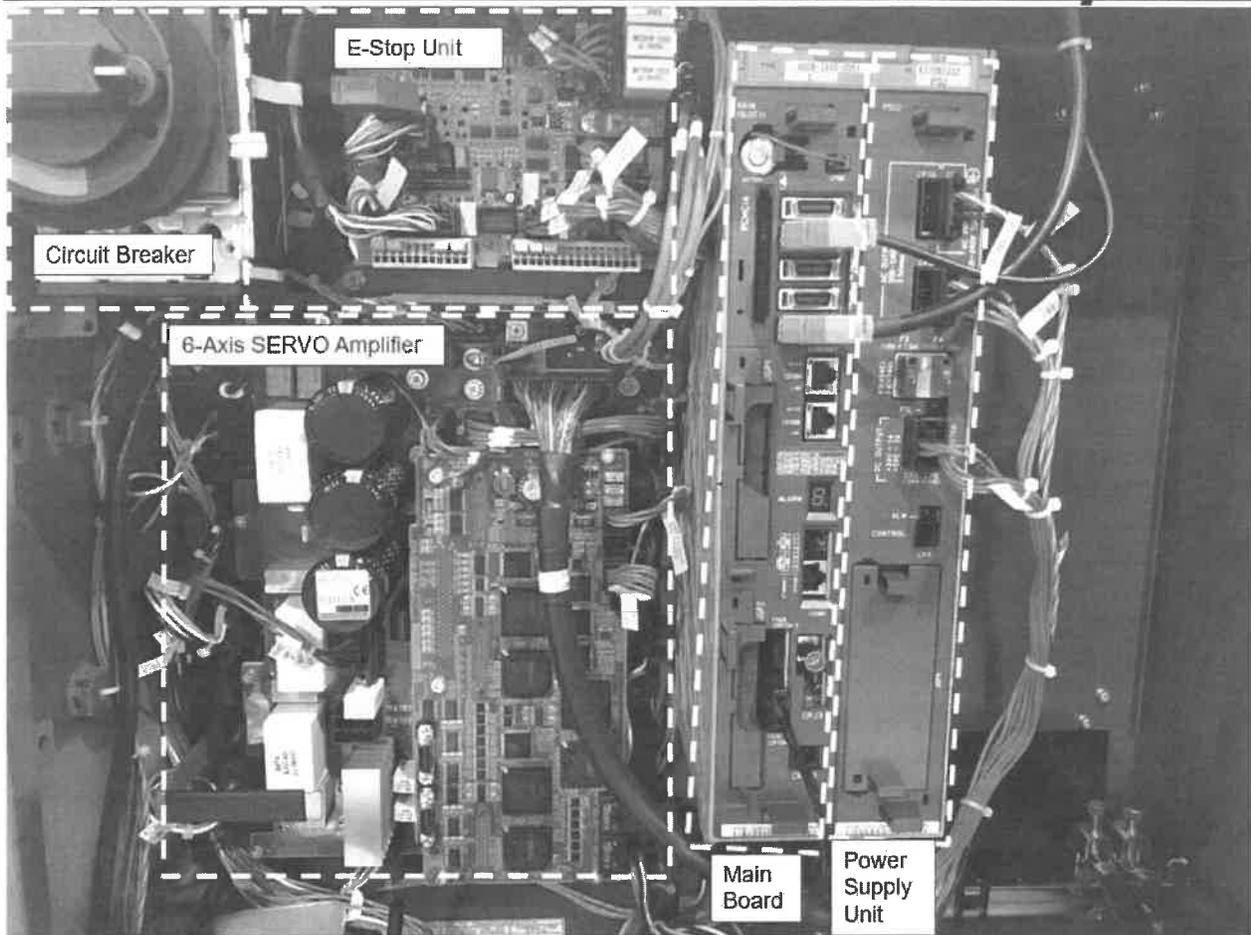
The controller contains the power supply, operator controls, control circuitry, and memory that direct the operation and motion of the robot and communication with external devices. You control the robot using an iPendant or an operator panel.

- Some systems contain an optional user operator panel (UOP) that provides a remote user interface to the controller.
- The motion system directs robot motion for all robot axes, including any extended axes and up to seven additional motion groups.
- Controller memory stores the software in addition to any user-defined programs and data.

There are three common styles of controller:

- A-Size Controller
- B-Size Controller
- Mate Controller

Regardless of the style of cabinet the controller houses all of the necessary items to operate the standard six axis robot and communicate externally. In some cases there may be an Auxiliary Control Panel (ACP) attached to the main controller. This houses any extra hardware that is required for the robot system to operate. An example would be an additional servo amplifier for an extended axis or additional motion group.



- **Circuit Breaker:** If the electric system in the controller malfunctions, or if abnormal input power causes high current in the system, the input power is connected to the circuit breaker to protect the equipment.
- **E-stop unit:** This unit controls the emergency stop system of the robot controller. It also has user interface terminals of safety relevant signals, external on/off signals etc.
- **Power supply unit:** The power supply unit converts the AC power to various levels of DC power.
- **Main board:** The main board contains a microprocessor, its peripheral circuits, memory, and operator's panel control circuit. The main CPU controls servo mechanism positioning.
- **6-Axis Servo amplifier:** The servo amplifier controls servomotor, Pulsecoder signal, brake control, EE Connector, overtravel, and hand broken.

WARNING

Lethal voltage is always present at the input side of the circuit breaker whenever the controller is connected to a power source. Be extremely careful to avoid electrical shock. DO NOT perform controller maintenance without proper training.

4: Robot System

4.4.1 Motion Groups

A motion group is a group of axes that work together to perform motion. As compared to a person: your shoulder, elbow, forearm, wrist, and fingers all work together to move your arm. In robot terms, these items would all be combined as one motion group.

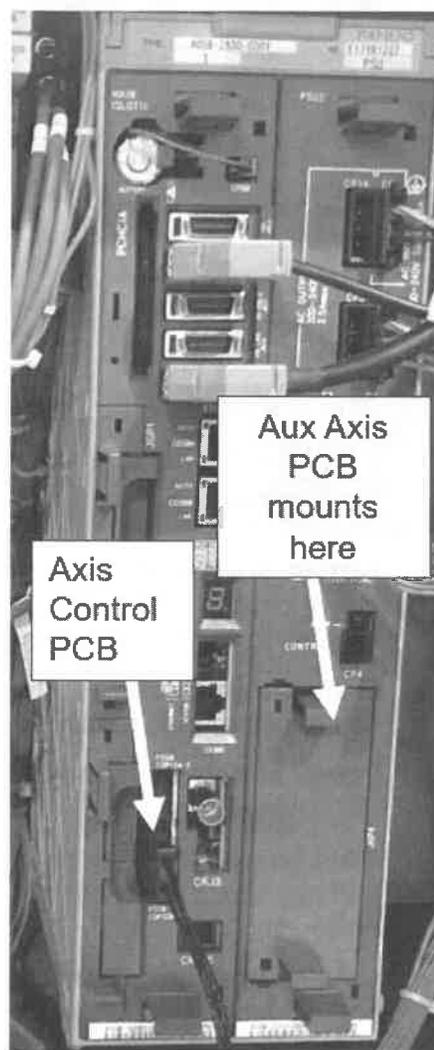
The controller optionally allows you to create up to eight motion groups. By default, the first motion group is a robot arm. Additional motion groups can be set up to perform tasks that are executed simultaneously with those of the robot.

Extended axes are the available axes controlled by the controller beyond the standard number of robot axes. Depending on your system setup, many applications use extended axes on jobs that require a rotary axis, sliding axis, rail tracking, opening devices, or line tracking.

Each motion group can contain up to a maximum of nine axes. The total number of axes cannot exceed 72. Motion groups are independent, but a maximum of two motion groups can perform Cartesian coordinated linear or circular interpolated motion within a single program.

The controller can support:

- Up to four full (9 axes) kinematics devices (robot mechanical units)
- Up to eight motion groups
- Non-robot motion groups of one to four axes can be defined.
- Up to 72 axes
 - Up to 36 axes can be supported on the Axis Control PCB*.
 - If needed, up to 24 axes can be supported on an Auxiliary Axis PCB. Only 2 auxiliary axis PCBs can be supported.*
- One to three extended axes can be added to a motion group.
 - The extended axes cannot be used independently of the motion group.
 - Each extended axis adds a position data field (E1, E2, E3) to the motion group data.
- You cannot exceed nine axes per robot motion group; you cannot exceed four axes per non-robot motion group.



*  **NOTE:** Refer to the Software Installation Manual Appendix G for information on axis setup

 **NOTE:** Supporting more than six axes will require additional hardware/software.

4: Robot System

HandlingTool

4.4.2 Software

The FANUC software works in conjunction with the robot and the 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 HandlingTool Application software loaded into the robot controller:

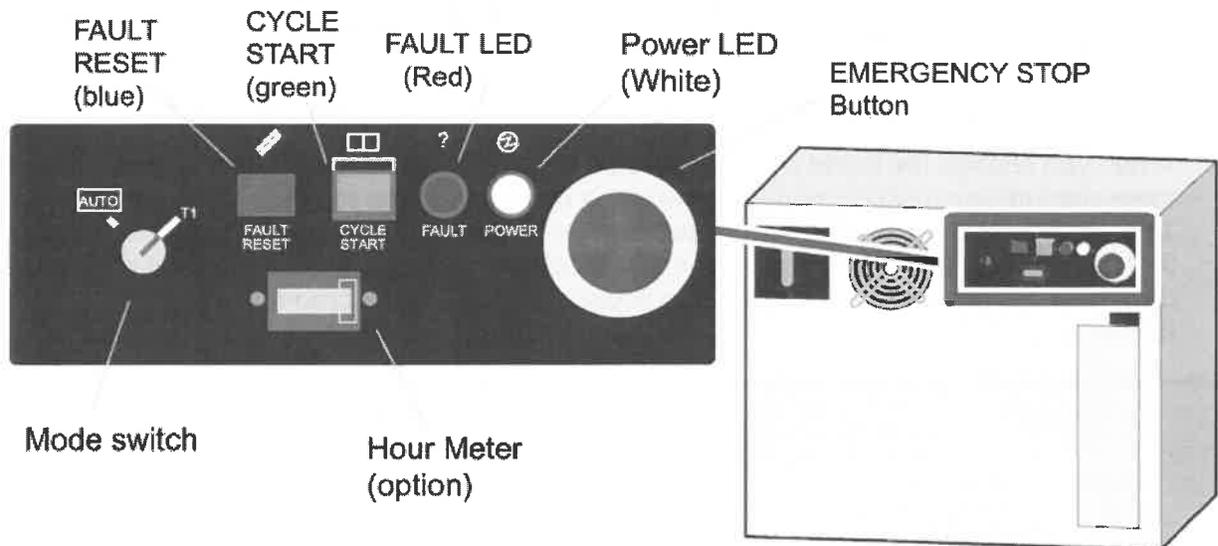
- Defines the function of the robot.
- Contains specific fault isolation and diagnostics.
- Defines communication protocols.

There are two versions of material handling software:

- LR HandlingTool (lighter version application – installed on all LR Mate robots only)
- HandlingTool for all other material handling robots

4.5 Standard Operator Panel

The Standard Operator Panel (SOP) allows control of certain basic robot functions and provides indication concerning robot operation. The SOP contains buttons, key switches, and connector ports and is part of the controller.



4: Robot System

Item	Description
Emergency Stop	Applies Robot Brakes and Removes power from motors
Mode Switch	Selects either Auto, T1, or T2 mode
Fault Reset	Clears a fault message from the iPendant Screen after being corrected
Cycle Start	Can start programs in Auto Mode. When lit indicates that a program is running.
Fault Indicator	Indicates that a fault has occurred
Power Indicator	When lit indicates the controller is on.
Hour Meter	Tracks Servo On time

NOTE: The Mate Controller SOP only features the Mode Switch, Cycle Start, and Emergency Stop.

4.5.1 Mode Select Switch

The MODE SELECT switch is a key switch installed on the controller operator panel. 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.

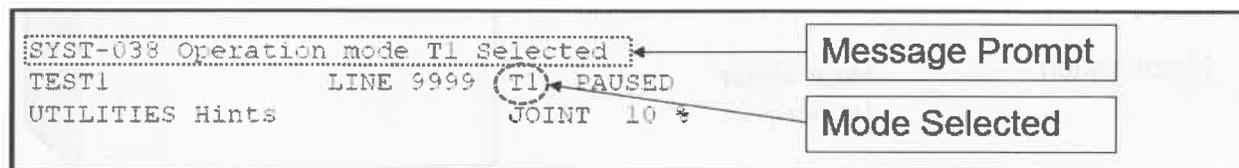


2 Mode Switch



3 Mode Switch

When you change the mode using the MODE SELECT switch, the robot is paused and a message indicating which mode is selected is displayed at the top of the iPendant screen. This message will be overwritten by status or error messages during operation. The mode that is currently selected is displayed immediately below the first message, and remains displayed. For example, if T1 mode is selected, you will see a screen similar to the following:



When choosing an operation mode ensure you have released the DEADMAN Switch. Changing modes with the DEADMAN Switch in the center position will generate a SYST-042 DEADMAN Defeated. To clear this fault, release the DEADMAN Switch and press reset.

The operation modes that can be selected using the Mode Select switch are described below:

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

- Program Activation:** Programs can only be activated from an enabled iPendant with the DEADMAN Switch centered.
- 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 at 100% override, the robot's speed will be the program speed if the program speed is below the T1 mode safe speed. T1 mode safe speed is defined as 250 mm/sec for TCP and 10% of maximum joint speeds. If the program speed is above the T1 mode safe speed, robot motion will be executed at the T1 mode safe speed. With lower overrides, the robot speed is reduced proportionally according to the override setting.
- Safety Equipment:** Safety fence is bypassed.
- Locking the Mode:** Remove the key to lock in the mode.

T2 (100%): Test Mode 2

- Program Activation:** Programs can only be activated from an enabled iPendant with the DEADMAN Switch centered.
- 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%.
- Locking the Mode:** You cannot lock the switch in T2. You cannot remove the key from the switch in this mode.

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. When the switch is in AUTO mode, you cannot start programs using the iPendant.
- Robot Speed:** The robot is operated at the specified maximum speed as defined in the program relative to the general override.
- Safety Equipment:** The safety fence is monitored.
- Locking the Mode:** You can lock the switch in AUTO mode by removing the key from the switch.

4.6 Starting Up the Controller

Turning on the robot provides power to the robot and controller. There are a number of ways to start up the controller, of these the most common is a Cold Start.

CAUTION

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

4.6.1 Conducting a Cold Start

During a cold start the controller:

- Initializes changes to system variables
- Initializes changes to I/O Setup
- Displays the HMI screen upon successful startup

Procedure 4-1 Performing a Cold Start

- 1 Visually inspect the robot, controller, workcell, and surrounding area. During the inspection make sure all safeguards are in place and the work envelope is clear of personnel.

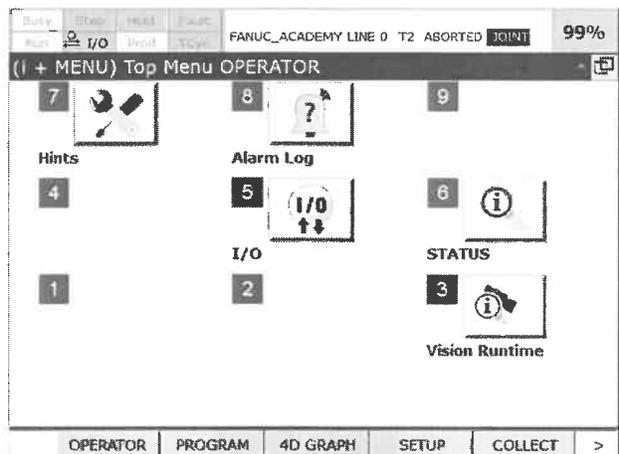
WARNING

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.

- 2 Turn the power disconnect circuit breaker or power switch on the operator panel to ON.
- 3 You should see a screen similar to the next page after a complete start up sequence.



R30iB Plus Home Screen



R30iB Home Screen

4.6.2 Cycle Power from a Powered Up Controller

During operations it may become necessary to cycle power on the controller. This can be caused from a number reasons. For example it may be necessary to clear a fault or to initialize a setting.

There are two ways to cycle power on the controller:

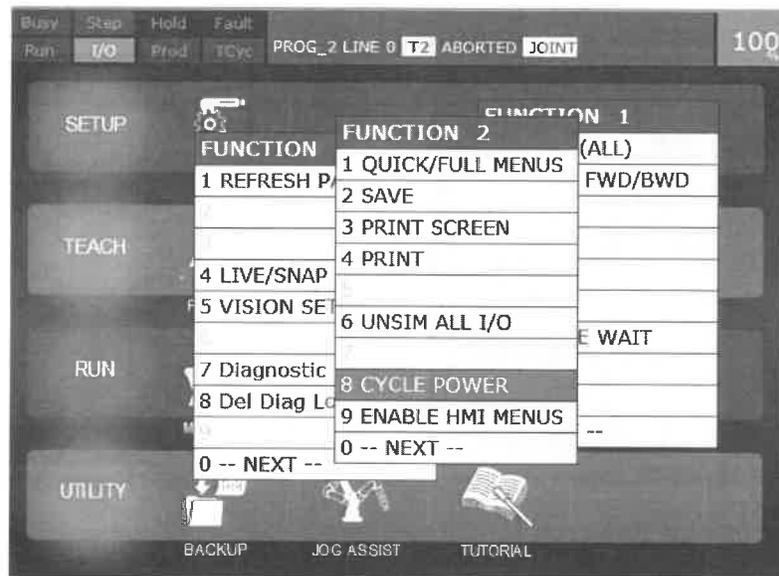
- Turn off the main breaker and turn it back on or
- Cycle power from the iPendant.

The safer route is to utilize the iPendant to cycle power. When the pendant executes the operation there is a built in delay to allow residual power to be released before reapplying power. This will minimize the likely hood of damaging any of the Printed Circuit Boards (PCB's) inside the controller.

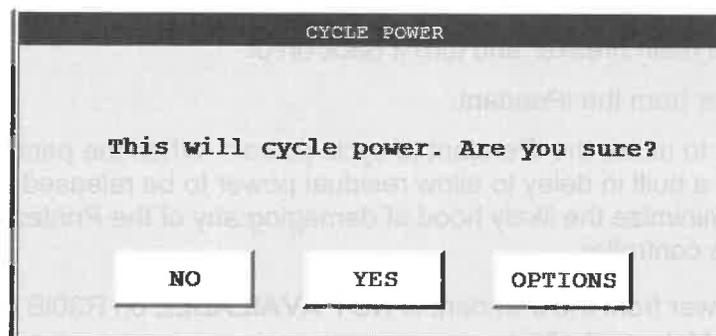
NOTE: Cycling power from the iPendant is **NOT AVAILABLE** on R30iB Plus Mate controllers and R30iB Mate controllers.

Procedure 4-2 Cycling Power from the iPendant

- 1 If motion is executing, press hold to pause the motion.
- 2 Enable the iPendant
- 3 Press Function
- 4 Press -0- Next
- 5 Select -8- Cycle Power



- 6 On the Prompt Menu select Cold
 - a On the next Prompt select yes



- 7 After a brief pause the system will shut down and then restart.

Procedure 4-3 Cycling Power from the Main Breaker

- 1 If motion is executing, press hold to pause the motion.
- 2 Press the ON / OFF button (if applicable)
- 3 Turn the breaker off
- 4 Count to five
- 5 Turn the breaker back on.

4.6.3 Other Start Up Methods

As previously stated there are a number of ways to power up the robot controller. Although a cold start is the most common, the other methods have specific uses. Refer the chart below for more information on other start up methods and the procedure to conduct them.

4: Robot System

HandlingTool

Start Method	Description	Procedure
Controlled start (START CTRL)	<ul style="list-style-type: none"> 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 iPendant screens and the current robot configuration Allows you to unsimulated all I/O Does not allow you to load iPendant programs 	<p>With the Controller ON</p> <ul style="list-style-type: none"> Press FCTN, -0- Next, and select -8- CYCLE POWER, On the prompt menu select CNTRL then verify YES. The controller will power up in the Controlled Start Mode <p>OR</p> <p>With the Controller OFF</p> <ul style="list-style-type: none"> Press and hold the PREV and NEXT keys Turn on the controller. On the configuration menu select Controlled Start.
Hot Start	<ul style="list-style-type: none"> Turns on power to the robot and controller without entering Boot Monitor. Hot start is useful when you are in production and an error occurs requiring controller restart. The controller will restart in the same condition it was prior to shutdown. 	<ul style="list-style-type: none"> With controller ON <ul style="list-style-type: none"> Press MENU > SYSTEM > Config. Setup item 1. Hot Start to TRUE. Cycle power on the controller.

4.7 Powering Down the Controller

When powering down the FANUC controller there are no special considerations needed to save information to robot memory. As changes are made they are automatically saved to internal robot memory. However, if you power down the robot controller while in executing a program power will be removed from the motors causing the brakes to engage, generating an emergency stop.

Procedure 4-4 Powering Down the Controller

- 1 If motion is executing, press hold to pause the motion.
- 2 Press the ON / OFF button (if applicable)
- 3 Turn the breaker off



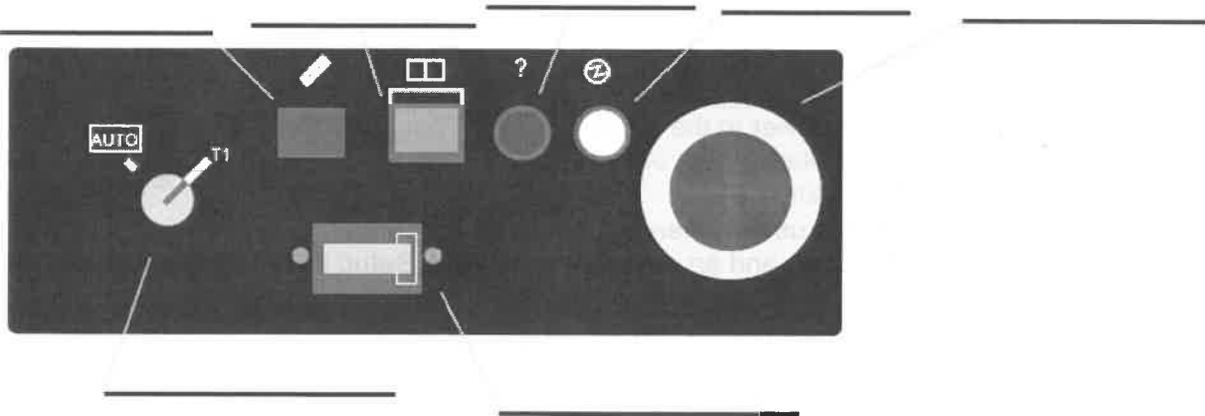
WARNING

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

4: Robot System

4.8 Chapter Review

1. List the major axes: _____, _____, and _____
2. List the minor axes: _____, _____, and _____
3. Identify the components of the SOP:



4. Which Mode limits all Cartesian speed to less than 250 mm/sec and joint speed to less than 10%?
A Auto Mode
B T2 Mode
C T1 Mode
D Neither, this is a system variable setting

Lab 1 (Optional) Robot System Introduction

Student Name: _____

Assignment: The student will:

- Receive an introduction to the robot components.
 - Power on the controller, cycle power, power off the controller.
-

Condition: Given a FANUC Robot and a controller

- Step:**
- 1 Gather around the instructor and a designated robot
 - 2 Receive an overview of the robot axis (major/minor) and an overview of the components inside the controller.
 - 3 Perform power up/power down:
 - a Perform a cold start
 - b Cycle power (from the pendant if available).
 - c Power down the controller.

Completed:

Instructor: _____

5 iPENDANT

5.1 Learning Objectives

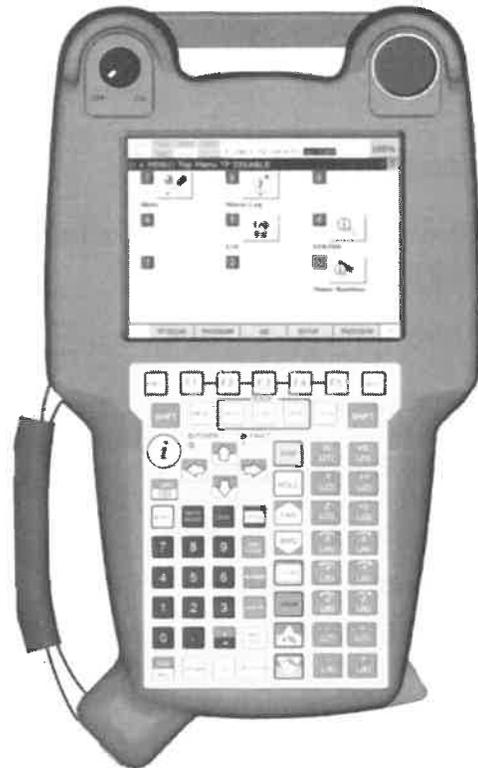
- Receive an iPendant Overview
- Learn the functions of the iPendant switches and keys
- Learn how to navigate the iPendant screens
- Perform iPendant Operations

5.2 iPendant Overview

The iPendant is the primary means with which to interact with the robot. The iPendant is the standard iPendant style that is included with the R30iB Plus or R30iB robot controller.

The iPendant provides:

- iPendant keys designed to make the FANUC software easy to use.
- An ON/OFF switch, Enabling Devices (DEADMAN switches), and an EMERGENCY STOP button.
- Popup menus.
- Multiple screens displayed simultaneously.
- A USB Port.
- A color graphics interface.
- Customized displays.
- A 3D graphics processing engine to display the robot, tooling, parts and other cell components in 3D desktop quality graphics.
- User-customized help and web pages.
- Integrated Help and Diagnostics
- Haptic features (option).
- A touch screen interface (option).
- Internet/intranet access (option).



R30iB Plus Specific Features:

- Tutorials to teach how to perform basic functions; Jogging, creating programs, etc.
- iHMI Guides for setting up your robot

5.3 iPendant Switches and Keys

The iPendant includes switches that allows the user to energize the robot arm and enable the iPendant. There are also a numbers of keys that are used to perform a number of functions within the software such as:

- Display Menus
- Move the Robot
- Select menu items
- Perform Specific Functions
- Assist in programming

5.3.1 ON / OFF Switch (ENABLE / DISABLE Switch)

The iPendant can either be enabled or disabled utilizing the on/off switch that can be found either on the top left of the pendant. The status of the pendant (enabled/disabled) dictates the functions that are available to the operator.



iPendant On (Enabled):

- Gives motion control solely to the operator
- Ability to manually move the robot
- Create and edit programs
- Test / Run programs
- Perform manual functions

iPendant Off (Disabled):

- Set up production operations
- Configure the application
- Edit programs (background edit only)
- Check Status
- Monitor Production

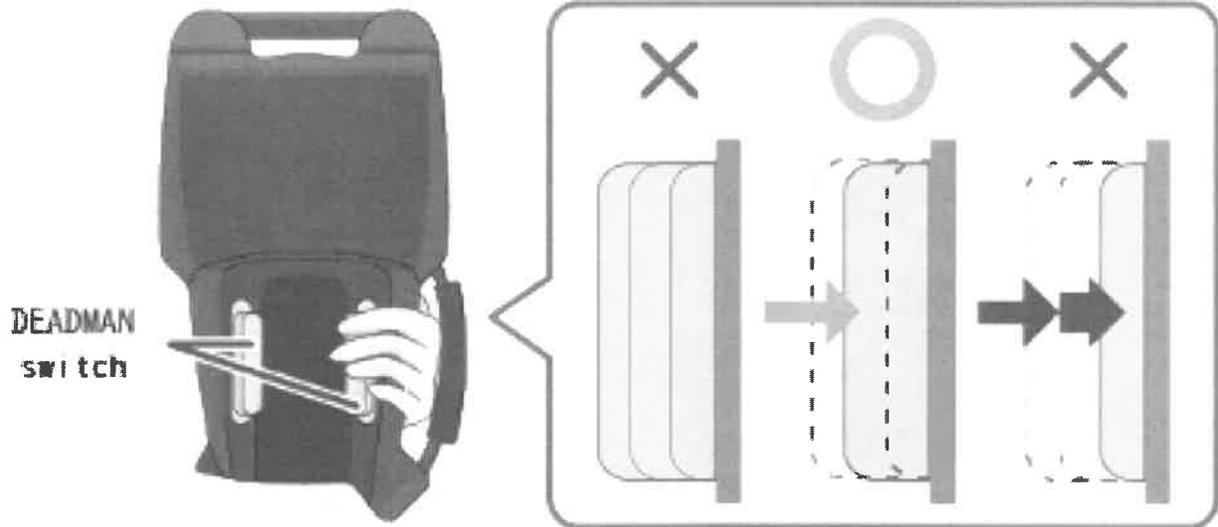
5.3.2 Deadman Switch

Two DEADMAN switches are located on the back of the iPendant, and are used as enabling devices. Each DEADMAN switch is a three-position switch. When the iPendant is enabled, this switch allows robot motion only while a DEADMAN switch is gripped in the middle position.

If the iPendant enabling switch were in the ON position and an operator were to release their grasp of the DEADMAN switch, servo power is removed from the controller and the robot brakes are immediately applied.

- If the DEADMAN switch is completely gripped or is released, **no motion is allowed**.
- Press the DEADMAN switch in the middle position to run the robot.

- At least one DEADMAN switch must be held in the center position for the duration of manual motion (program testing / jogging).



5.3.3 Emergency Stop Button

The red EMERGENCY STOP button is located on the front of the iPendant in the upper right hand corner. You can press the button to stop the robot immediately in case of accident or failure.

The EMERGENCY STOP Button will function whether the iPendant is on or off. To disengage the EMERGENCY STOP rotate the button clockwise. The EMERGENCY STOP button should only be pressed in the event of an emergency.

In a non-emergency situation it is better for the mechanical unit to utilize the HOLD key. The HOLD key will stop the robot in a controlled manner, generating less stress on the mechanical unit.

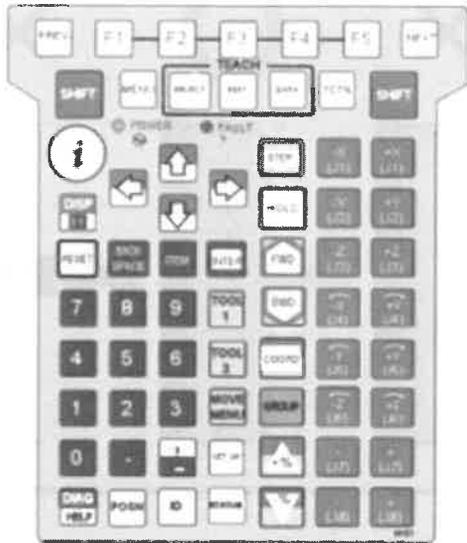
NOTE: This section in no way is inferring that the EMERGENCY STOP cannot and should not be used.

CAUTION

Under certain load/speed conditions it could be possible to damage some mechanical units when E-stop button is pressed. In non-emergency situations the HOLD button is the preferred method of stopping the robot.

5.3.4 iPendant Keys

The iPendant key pad is comprised of a series of keys that are designed to make the pendant operations and navigation easier and more user friendly.



Standard Material Handling Keypad

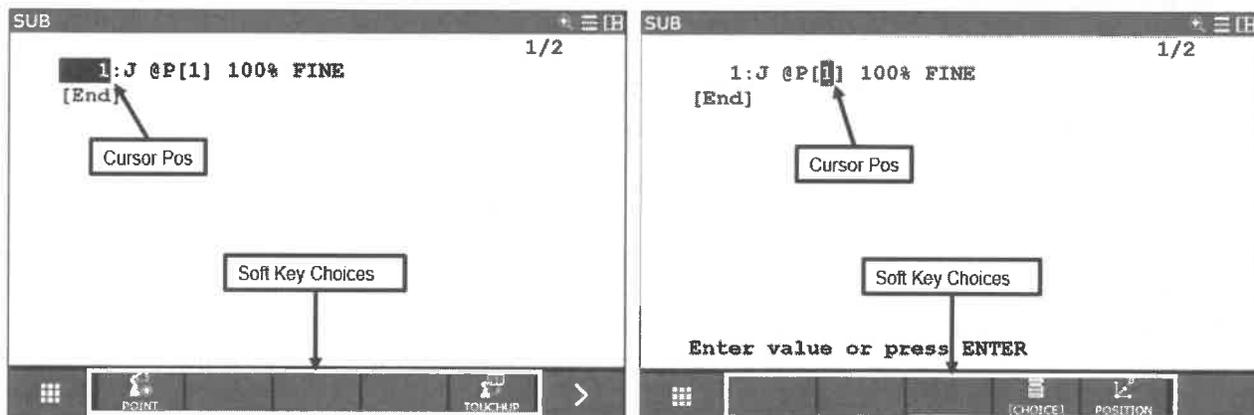


Global Symbol Keypad

5.3.4.1 Soft Keys

The iPendant keys are further broken down between hard keys and soft keys. There are five soft keys on the iPendant, they are keys labeled F1-F5 (function keys). Every other key on the pendant is designated as a hard key, as their functions do not change.

What makes a key a soft key is the functionality of the key. On the iPendant depending on the screen displayed and in some cases the position of the cursor on the screen the function of the soft key will change. Therefore is vital as an operator that you are aware of function of the F1-F5 keys before pressing. The function of these keys are displayed along the bottom of the iPendant above the function keys.



5.3.4..2 iPendant Key Sub Groups

The iPendant keys are divided into the following groups

Navigation and Data Entry Keys

The F1 through F5 function keys are used to make choices based on the teach pendant display. Each function key has a unique label depending on the menu displayed on the teach pendant screen.

The diagram shows the iPendant keypad with callouts for the following keys:

- PREV**: The PREV key restores the most recent state. In some cases, the screen might not return to the immediately preceding status.
- MENU**: The MENU key is used to display the screen menu.
- i key**: The *i* key is a special key. When you press the *i* key together with other keys, the special screen is displayed. When you press *i* + HELP, the Help for *i* key screen is displayed.
- F1-F5**: The F1 through F5 function keys are used to make choices based on the teach pendant display. Each function key has a unique label depending on the menu displayed on the teach pendant screen.
- NEXT**: The next page key is used to display the next set of function keys.
- FCTN**: The FCTN key is used to display the function menu.
- SELECT**: The SELECT key is used to display the program selection screen.
- EDIT**: The EDIT key is used to display the program edit screen.
- DATA**: The DATA key is used to display the program data screen.

Robot Motion Keys

The diagram shows the iPendant keypad with callouts for the following keys:

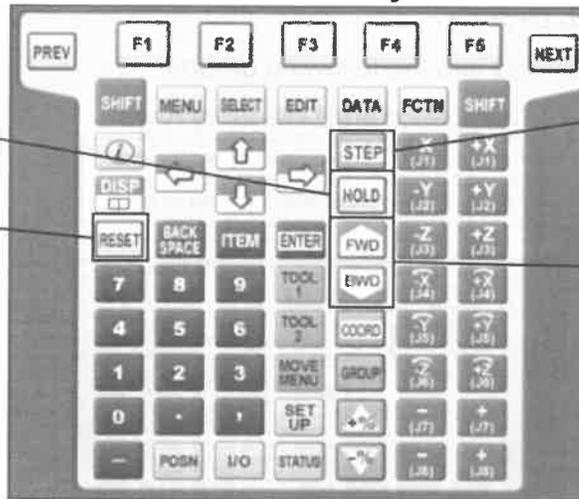
- COORD**: The COORD key selects a jog coordinate system. When this key is pressed while a SHIFT key is pressed and held down, a jog menu for changing the coordinate system is displayed.
- GROUP**: The GROUP key is used to switch groups.
- Override**: The override key adjusts the feedrate override.
- SHIFT**: The SHIFT key is used to jog the robot, teach the position data, and start a program. The right and left SHIFT keys have the same function.
- Jog Keys**: The jog keys are effective while a SHIFT key is pressed. They are used to jog the robot.
- Extended Axes**: These keys are used to jog extended axes or servo gun axes..

5: iPendant

Execution Keys

The HOLD key causes a program to halt

The RESET key is used to clear an alarm.



The STEP key selects step or continuous test operation.

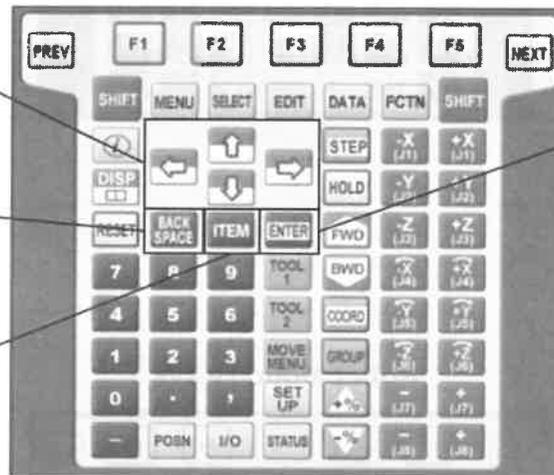
The FWD key or BWD key (+ SHIFT key) starts a program. When the SHIFT key is released during regeneration, the program halts.

Editing Keys

The arrow keys are used to highlight or select an item on the screen.

The BACK SPACE key deletes the character or numeral immediately before the cursor.

The ITEM key moves the cursor to a line whose number is specified.



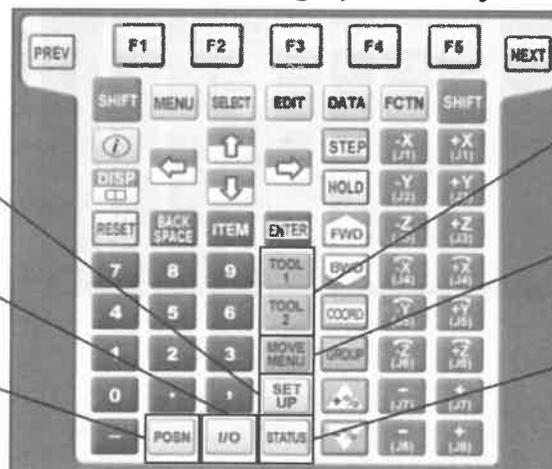
The ENTER key is used to process and activate the current information set.

Material Handling Specific Keys

The SET UP key displays the SETUP screen.

The I/O key displays the I/O screen.

The POSN key displays position data.



The TOOL 1 or 2 key displays the tool 1 or 2 screen.

The MOVE MENU key is not supported for HandlingTool.

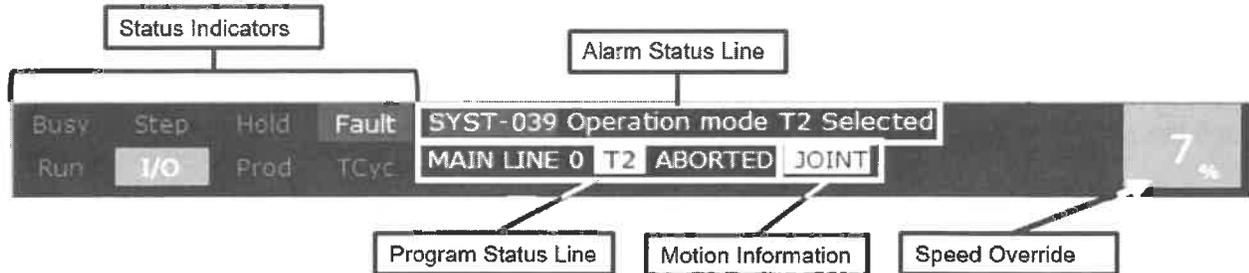
The STATUS key displays the STATUS screen.

5: iPendant

Handling Tool

5.4 Status Bar Displays

The status bar displays dynamic controller and robot status information.



Status Item	Description
Status indicators	<p>The iPendant status indicators indicate the system condition when you are using the iPendant to control the system.</p> <ul style="list-style-type: none"> • Busy indicates that the controller is processing information. • Step indicates that the robot is in step mode. • HOLD indicates that the robot is in a hold condition. HOLD is not on continuously during a hold condition. • FAULT indicates that a fault condition has occurred. • Run indicates that a program is being executed. • The last three indicators are application-specific
Alarm Status Line	<p>This item displays the current alarm. Touching the Alarm or Program Status Line will be the same as pressing the DIAG key, which can be used to display diagnostics for the current alarm.</p>
Program Status Line	<p>This item indicates the name of the currently executing program, and the current line number and status. It also indicates the mode AUTO, T1, or T2.</p>
Motion Information	<p>This item lists the following information:</p> <ul style="list-style-type: none"> • Coordinate System • Group • Subgroup
Speed Override	<p>The speed override is a percentage of the maximum speed at which the robot will move. A speed override of 100% indicates that the robot will move with the maximum possible speed</p>

5.4.1 LED Indicators

The Two LED Indicators on the iPendant are Power and Fault.

- **Power:** Indicates controller power is on.
- **Fault:** Indicates that a fault has occurred.



5.5 Screen Navigation

There are a number of ways to navigate the iPendant screen.

5.5.1 Touch Screen Feature

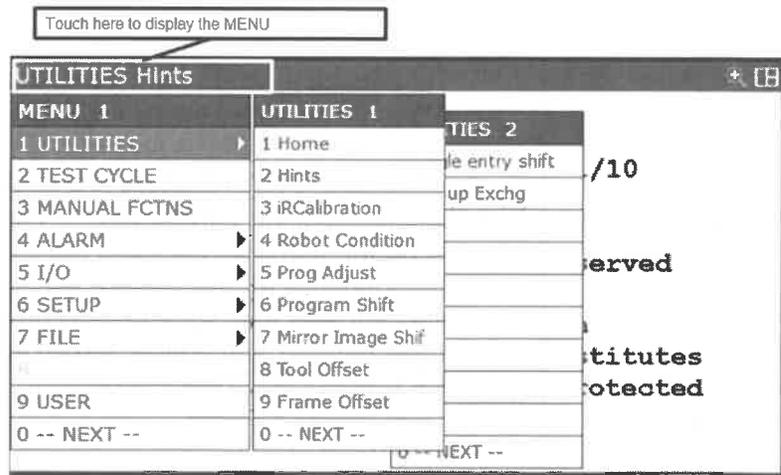
If you have the optional touch screen available on your iPendant, you can select items by touching them on the screen. The item is activated as soon as you release the touch. You can touch the screen anywhere to focus the window when the screen is split.

NOTE: Prompt box and Warn windows will not allow touch and release events. These events are typically shown in yellow on the iPendant screen.

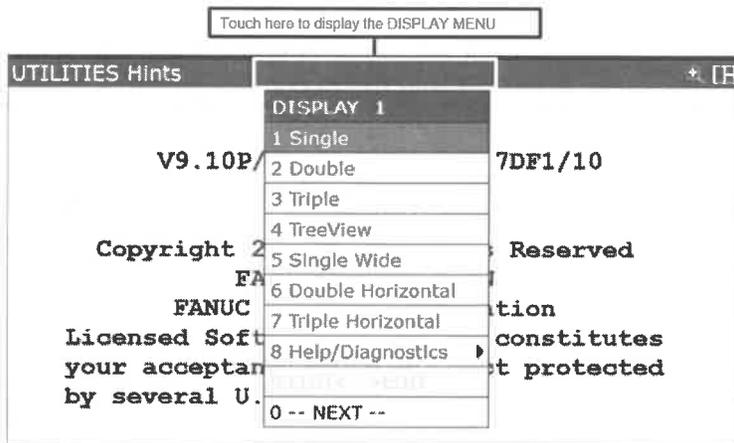
5.5.2 Display Shortcuts

The title bar of the focused window is divided into three parts and can be touched to activate a shortcut to the MENU, DISPLAY, and FCTN menus.

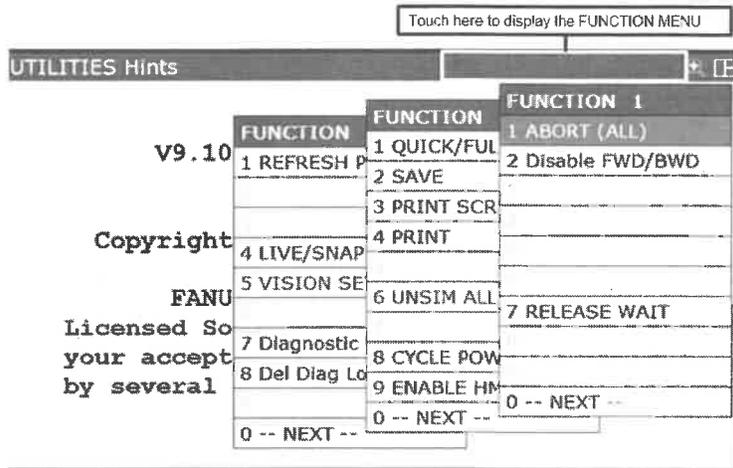
- MENU



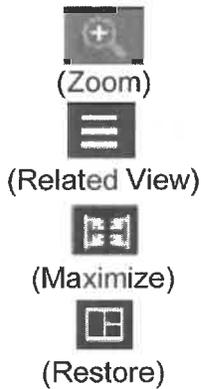
- DISPLAY



- FUNCTION



The title bar may also be equipped with additional indicators such as zoom, related view, or maximize/restore, you can touch the icon to activate the feature.

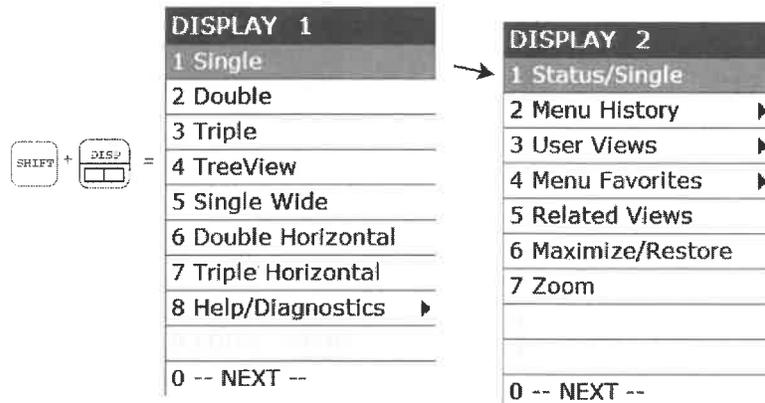


- Allows the user to zoom in on a single screen, you can only zoom in once, depress this icon again to return to original view.
- The Related View Menu displays other menu choices that are available. It can display Related Views or provide a shortcut to the related iHMI Guide if one is available.
- With a split screen configuration this item allows to make the window selected displayed in single window.
- Reverts the display back to split screen selected prior to selecting maximize.

5.5.3 Split Screen

Information can be displayed in multiple windows. You can activate the Display menu by pressing the SHIFT key and the DISP key (the Display key) at the same time.

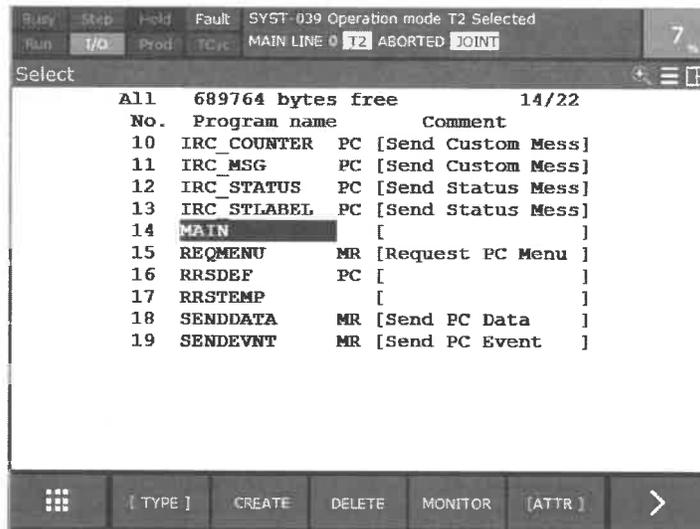
To select a window mode, you can either use the arrow keys to highlight a mode and then press ENTER, or you can type the number of the mode you want.



The table below shows some of the options available in split screen mode:

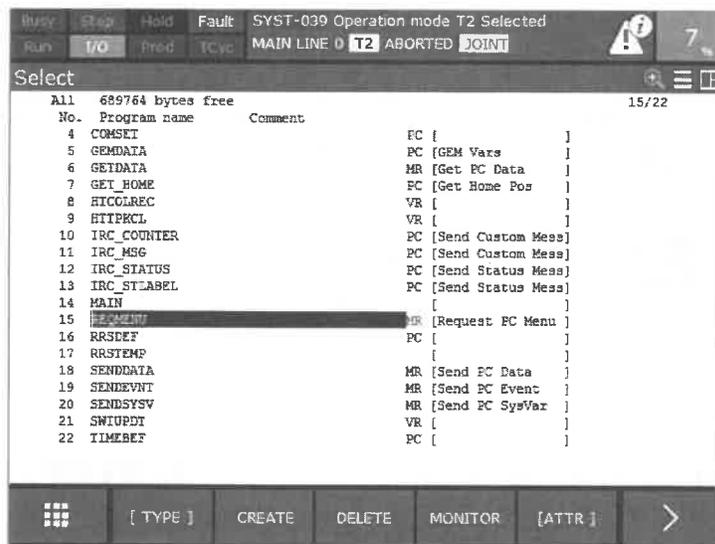
Single Window:

One window is displayed



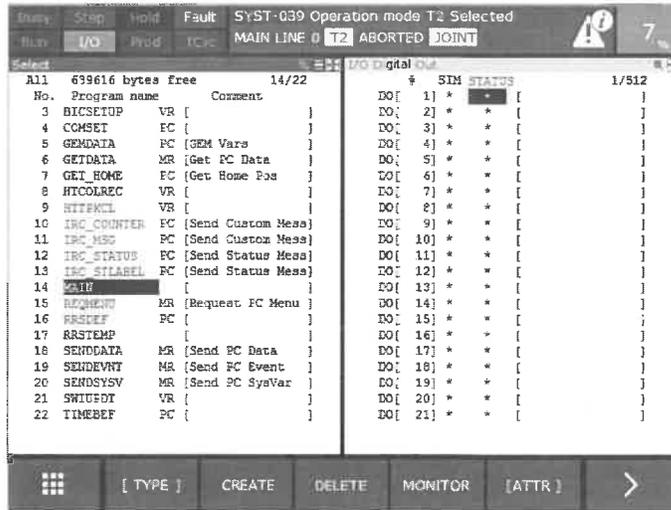
Single Wide:

One wide screen window is displayed.



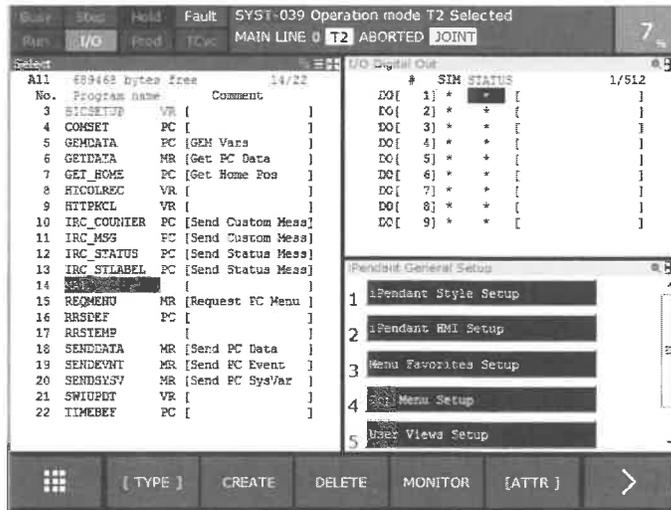
Double Window:

Two different windows can be displayed at one time.



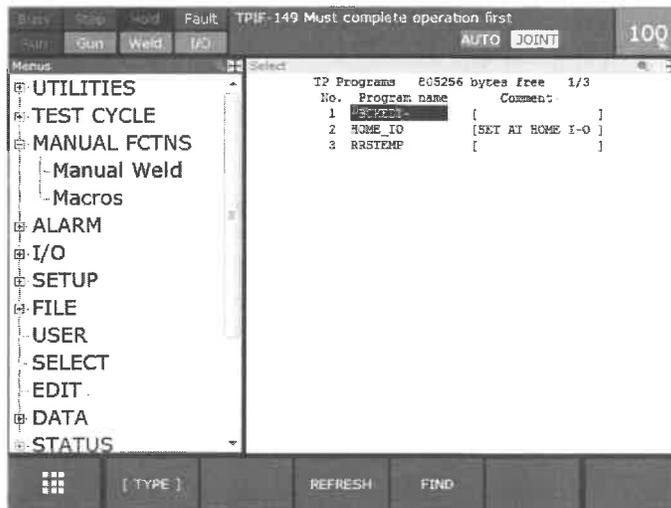
Triple Window:

Three different windows can be displayed at one time.



Tree View:

The Tree View window presents a hierarchical view of information. Each item (often called a branch or a node) can have a number of sub items. An item can be expanded to reveal sub items, if any exist, and collapsed to hide sub items.



5: iPendant

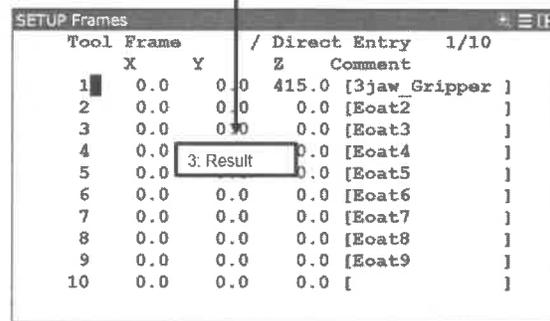
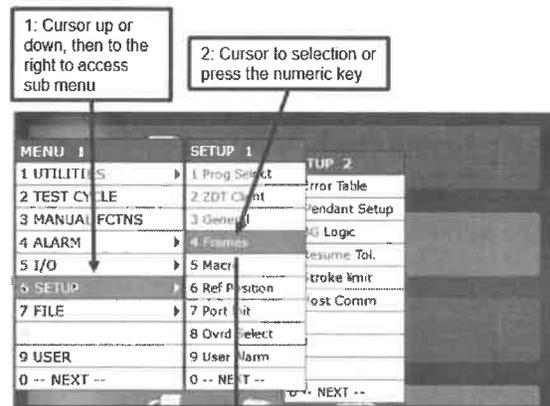
5.5.4 Menu Navigation

The menus associated with the FANUC software feature fly out menus. This setup allows the user to preview the sub items within each top level menu if applicable.

When displaying either the regular MENU items or the FUNCTION MENU items there are two ways to navigate the selections; using the arrow keys or the numerical keypad.

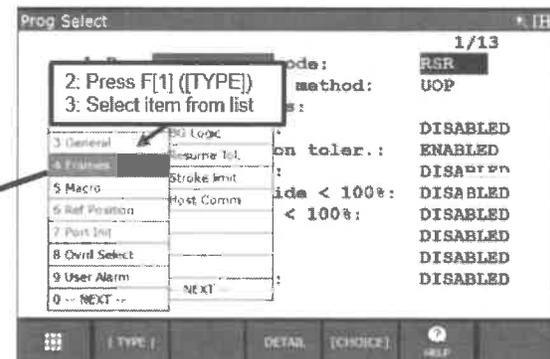
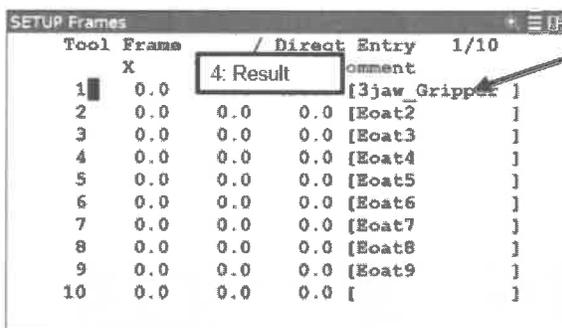
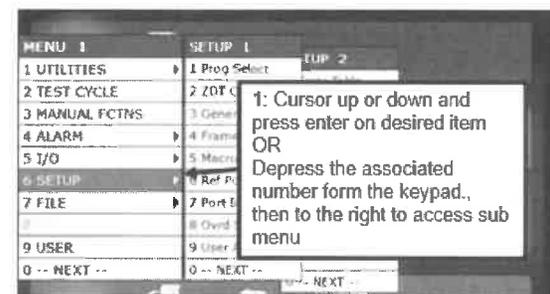
Procedure 5-1 Navigating the Menu

- 1 Press MENU
- 2 Cursor up or down to your desired MENU, notice the items with the block arrow fly out to display the sub items.
- 3 Cursor to the right on the desired item to access the sub menu item
- 4 Cursor to the item, or press the associated number from the keypad.



Procedure 5-2 Navigating the Menu with Soft Keys

- 1 Press Menu
- 2 Cursor up or down to your desired item and press enter, or depress the associated number from the keypad.
- 3 Select F[1] ([TYPE])
- 4 Choose your desired item from the submenu.



5: iPendant

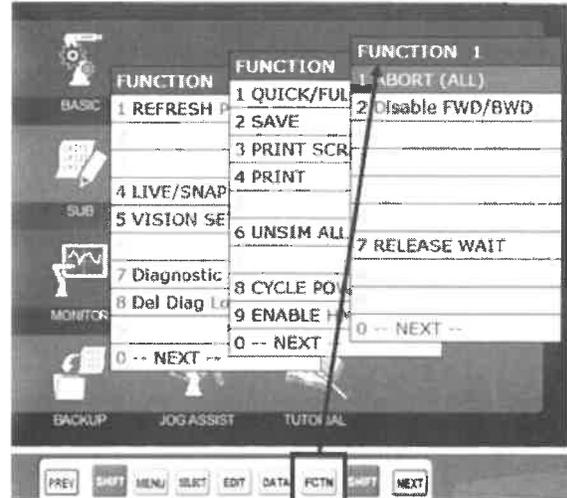
HandlingTool

5.5.5 Function Menu

The Function Menu gives the user access to a supplementary menu and items that assist with testing, setup, and saving specific information.

A number of these items require some form of prior setup in order to function properly. These items will be covered in later sections within this manual.

The items listed on the Function Menu are option and setup specific, your system may have more or less than shown here.



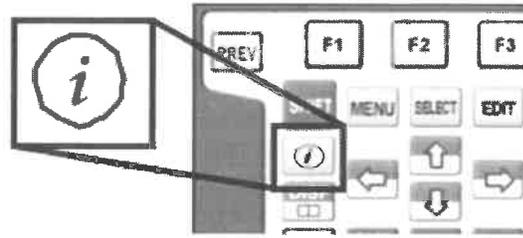
The below table lists the items that may be found on the Function Menu:

Item	Description
ABORT ALL	This item aborts all running or paused programs.
Disable FWD/BWD	This item allows you to disable the ability to execute program instructions when the SHIFT and FWD keys or SHIFT and BWD keys are pressed. The ability to use SHIFT FWD and SHIFT BWD will be disabled until you press FCTN and select DISABLE FWD/BWD again.
Release Wait	During program execution, wait release 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.
Quick / Full Menu	This item toggles the menu display between a full or a shorter menu list.
Save	This item saves the current program.
Print Screen	This item saves the current iPendant screen image to the default device as tpscrn.ls.
Print	This item saves the current program to the default device as program_name.ls.
Unsim All I/O	This item unsimulates all simulated signals.
Cycle Power	This option is available on the FCTN menu after a Cold start. This item toggles the power relay via external I/O. The iPendant must be enabled for this to work. If the iPendant is enabled, a prompt box will be displayed asking if you are sure. This is not available on an external robot connection — only from the iPendant.
Enable / Disable HMI Menus	This item enables (or disables) the optional HMI Menus in both FULL and QUICK Menu mode.
Refresh Pane	This item refreshes the page in the active window.

5: iPendant

5.5.6 *i* Key

The *i* key is a special key. When you press the *i* key together with other keys, a special screen is displayed. When you press *i* + HELP, the Help screen for the *i* key is displayed.



The table below describes the different views available with the *i* key.

Item		Function
	+ MENU	Displays the Top Menu
	+ SELECT	Displays the Select menu in the left pane and the 4D node map in the right pane for the program which is highlighted by the cursor.
	+ EDIT	Displays the TPP editor in the left pane and 4D node map in the right pane.
	+ DATA	Displays the position register data in the left pane and a 4D display of position registers in the right pane.
	+ FCTN	Displays the Related View Menu if one is available.
	+ POSN	Presents a 4D display of the robot using the last selected view.
	+ JOG KEY	Shows you the direction that the robot will move when the jog key is pressed when the 4D screen is displayed.
	+ DISP	Switches the display to the next window if one is available.
	+ COORD	Toggles jog indicators on and off.

5.6 iPendant Functions

There are many operations that the iPendant is capable of performing. Many of these items are designed to make the software easier to understand, provide shortcuts to menu items, or provide the user additional assistance in understanding the software.

5: iPendant

HandlingTool

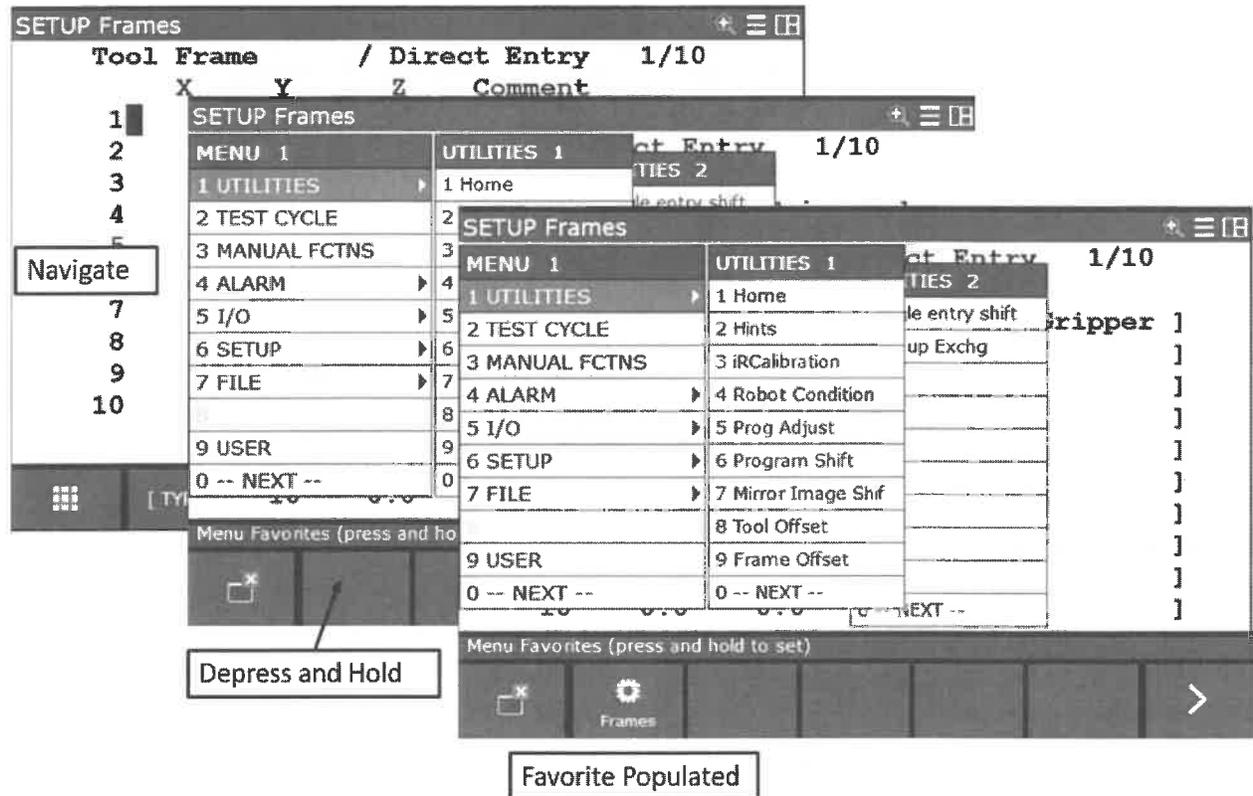
5.6.1 Menu Favorites

Menu Favorites allows you to generate and store a list of menus that you display most often. You can store up to ten favorites for quick access.

There are two ways to add a menu favorite; using the Menu dialogue box or using the DISPLAY Menu.

Procedure 5-3 Creating/Adding a Menu Favorite

- 1 Navigate to the menu item that you would like to make a favorite.
- 2 Press Menu.
- 3 In the dialogue box at the bottom of the screen depress one of the tiles for three seconds. The displayed menu name will be added to the box with an icon.

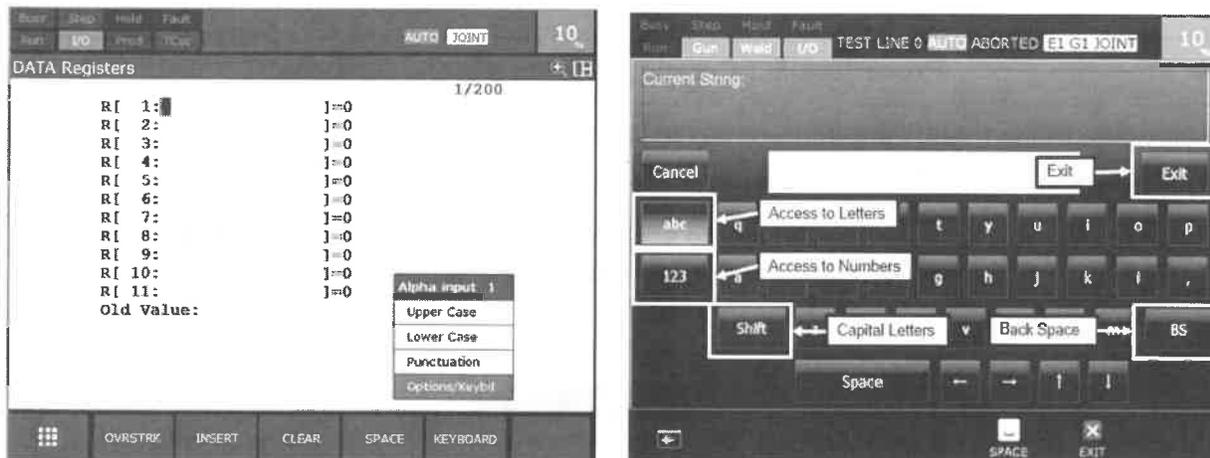


NOTE: You can override an existing favorite repeating the above procedure.

5: iPendant

5.6.2 Software Keyboard

You can use the software keyboard to input a character string. In order to use the software keyboard, set the cursor to the item that you want to input a character string, then press ENTER. Set the cursor to the item Options in the menu, then press F5, KEYBOARD.



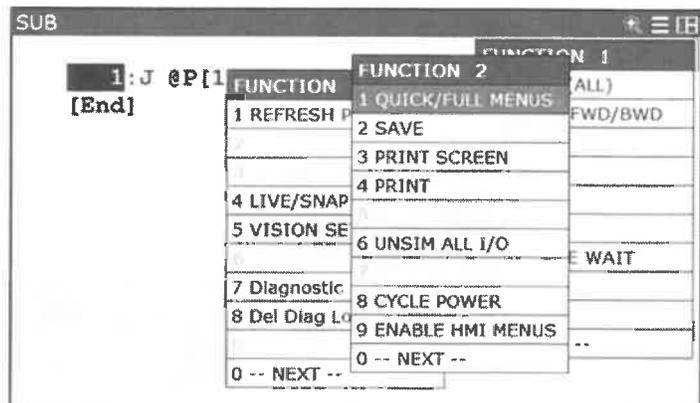
5.6.3 Quick / Full Menus

Using the Quick / Full Menu toggle allows the user to display a limited list of menu items as well as limiting the functions from the select and edit screens.

The Quick / Full Menu toggle is not password or write protected. It is merely a toggle bit. The procedure to set Quick Menus and revert back to Full Menus are the same.

Procedure 5-4 Toggling Quick / Full Menu

- 1 Press Function.
- 2 Press -0- Next.
- 3 Select -1- Quick / Full.
- 4 This will toggle from either Full Menus to Quick Menus and vice versa.



5: iPendant

HandlingTool

Select 688124 bytes free 20/23

MENU 1	UTILITIES 1	TIES 2	ant
1 MESSAGES	1 Home	1e entry shift	atus Mess]
2 TEST CYCLE	2 Hets	up Erchg	atus Mess]
3 MANUAL FCNTS	3 @Calibration		PC [
4 ALARM	4 Robot Condition		D] Event [
5 I/O	5 Prog Adjust		SysVar [
6 SETUP	6 Program Shift]]
7 FILE	7 Mirror Image SHF]]
8 USER	8 Tool Offset]]
9 USER	9 Frame Offset]]
0 -- NEXT --	0 -- NEXT --]]

[TYPE] CREATE DELETE MONITOR [ATTR] >

SUB 1/2

1:J @P[1] 100% FINE
[End]

TOUCHED >

Before Quick Menu

Select 688124 bytes free 20/23

QUICK 1	name	Comment
1 ALARM	S PC	[Send Status Mess]
2 UTILITIES	EL PC	[Send Status Mess]
3 TEST CYCLE	MR	[Request PC Mess]
4 DATA	MR	[Send PC Data]
5 MANUAL FCNTS	MR	[Send PC Event]
6 I/O	MR	[Send PC SysVar]
7 STATUS	[[]
	VR	[]
	PC	[]
9 USER	PC	[]
0 -- NEXT --		

TOUCHED >

SUB 1/2

1:J @P[1] 100% FINE
[End]

TOUCHED >

After Quick Menu

5.7 Chapter Review

1. **Enabling the iPendant gives motion control solely to whom?**
 - A The PLC
 - B The Controller
 - C The Operator (person with the pendant)
 - D Andy from accounting
2. **Which Position must the DEADMAN Switch be held to enable motion?**
 - A Released
 - B Centered
 - C Fully Depressed
 - D The iPendant does not have a DEADMAN Switch
3. **How many soft keys are there on the iPendant**
 - A 2
 - B 3
 - C 4
 - D 5
4. **Which Key Combination is needed to access the split screen display menu?**
 - A Shift + COORD
 - B Shift + Display
 - C Menu, next, Display
 - D You cannot split the screen on the iPendant
5. **Toggleing between Quick and Full Menus requires a password.**

True False

Lab 2 iPendant Operations

Student Name: _____

Assignment: The student will:

- Identify Soft Keys
- Navigate Menus
- Use the keyboard
- Split the iPendant Screen
- Set a Menu Favorite
- Toggle between Quick and Full Menus

Condition: Given a FANUC Robot and a controller

- Step: 1** Display the Select Screen. (See Section 5.3.4 iPendant Keys, pg. 42)
- a In the space provided write the items found in the soft key fields

						
--	--	--	--	--	--	--

- 2** Using the Menu, navigate to the Data top level menu and display the Registers submenu. (See Section 5.5.4 Menu Navigation, pg. 50)
- a From the register screen, utilizing the keyboard (See Section 5.6.2 Software Keyboard, pg. 54):
- b Name R[1] COUNTER and name R[2] OFFSET
- 3** Split the iPendant screen into either double view or triple view. (See Section 5.5.3 Split Screen, pg. 47)
- 4** In one of the newly displayed windows (not the left window), navigate to the top level menu of Setup and display the sub menu item Frames. (See Section 5.5.4 Menu Navigation, pg. 50)
- a Set the Frame screen as a Menu Favorite (See Section 5.6.1 Menu Favorites, pg. 53)
- 5** Toggle from Full Menus to Quick Menus. (See Section 5.6.3 Quick / Full Menus, pg. 54)

What is the benefit of turning on Quick Menus?

Completed:

Instructor: _____

6 JOGGING

6.1 Learning Objectives

- Jog the Robot in Joint and World
- View Positional Data

6.2 Jogging Overview

Jogging is manually moving the robot axes by pressing a sequence of keys on the iPendant. Before you add a motion instruction to a teach pendant program, you must first jog the robot to the position you want. There are a number of items to consider when manually moving the robot, chief among these are:

- Jog speed: How fast the robot moves when jogging
- Coordinate system: The way the robot moves when jogging

6.2.1 Jog Speed

The jog speed determines how fast the robot will move when the jog keys are pressed. 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 top right corner of every iPendant screen.

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 robot model. The maximum possible jog speed is defined by the tool center point (TCP) moving at and below 250 millimeters per second.

- A jog speed represented as a percent (%) will move the robot in continuous motion.
- A jog speed of FINE or VFINE indicates that the robot will move in incremental steps.

Care must be taken to ensure that the speed is set to a safe and acceptable level for jogging before the robot is moved.

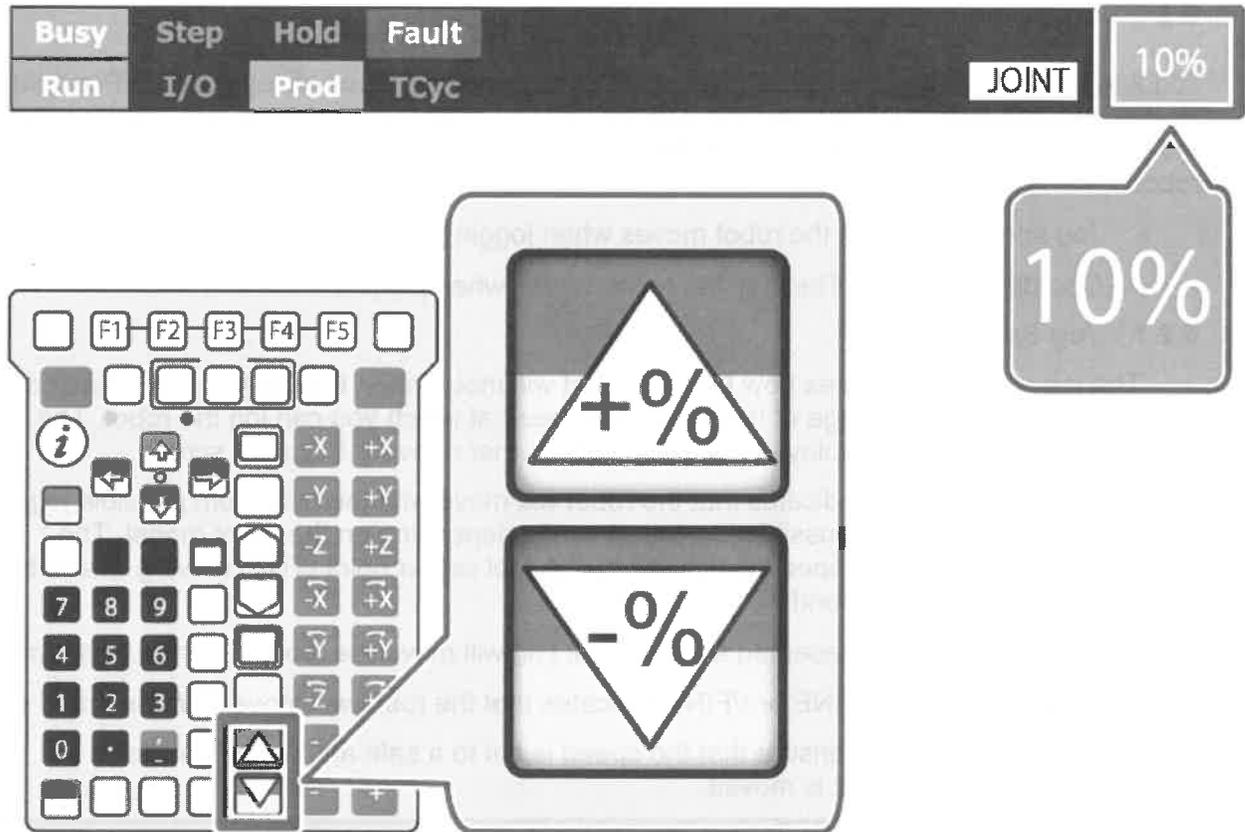
Speed Values	Joint	Cartesian
100, 95, 90, 85, ... 20, 15, 10, 5, 4, 3, 2, 1	% of jog speed	% of jog speed
FINE (incremental steps)	Approx. 0.001 degrees	Approx. 0.023 mm
VFINE (incremental steps)	Approx. 0.0001 degrees	Approx. 0.002 mm

 **NOTE:** When you use 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.

6: Jogging

Set the jog speed to a value that is appropriate for the conditions in the workcell, the kind 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.

Pressing the +% or -% keys will move the general override up or down in 5% increments by default. Holding the Shift key and +% or -% will increment/decrement the general override in steps of VFINE, Fine, 1%, 5%, 50%, 100%.



NOTE: If you are in T2 mode and you release the DEADMAN switch, the override speed will revert to a limited value such that the robot cannot move over 250mm/sec. This is to ensure safety when operation is resumed.

6.2.2 Coordinate System

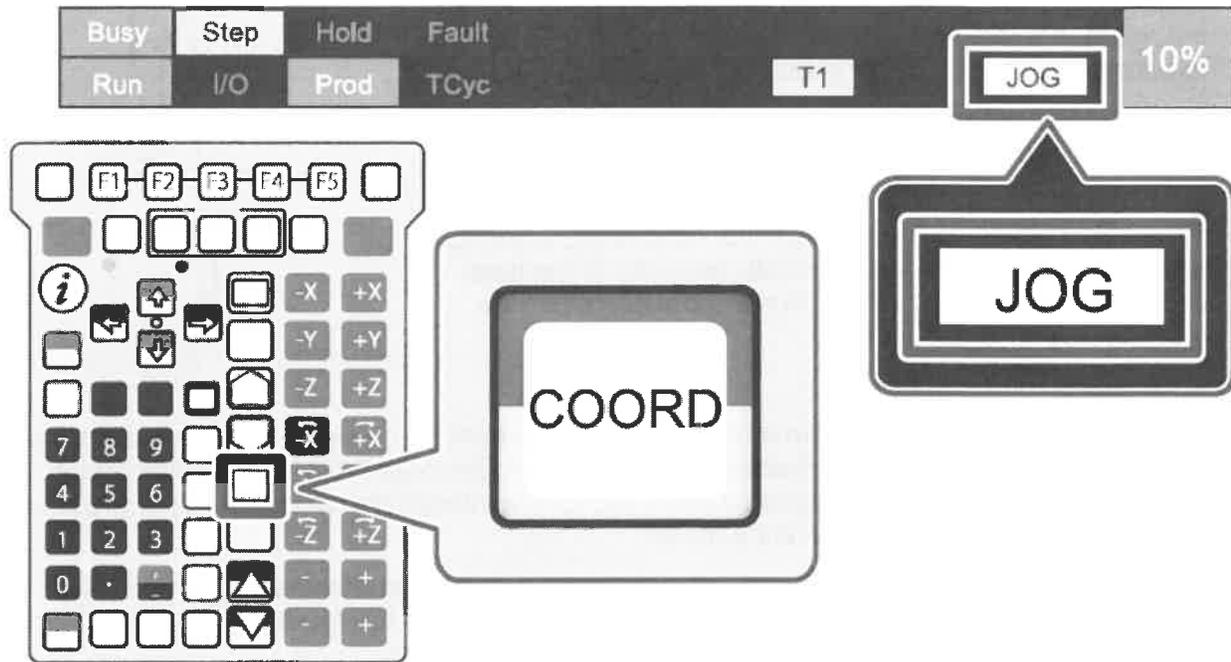
In jogging, a coordinate system defines how the robot will move. The coordinate systems are:

- Joint
- XYZ (includes World, User, Jog Frame)
- Tool

NOTE: XYZ, and Tool coordinate systems are commonly referred to as Cartesian Coordinate Systems.

6: Jogging

You change the coordinate system by pressing the COORD key on the iPendant. The coordinate system you choose is displayed in the upper right hand corner of the iPendant screen.



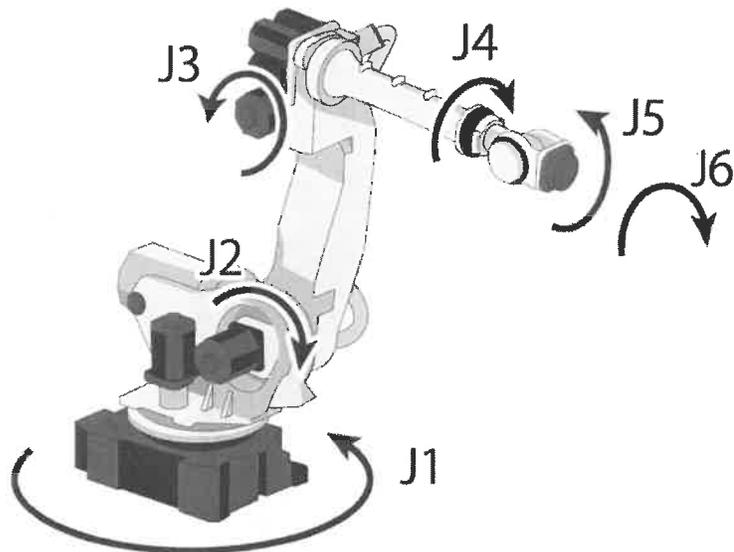
6.2.3 Jogging in Joint

The most efficient method of jogging is "JOINT" mode. This is the most efficient type motion because the processor does not have to calculate a linear path or maintain the tool center point movement along a Cartesian plane.

Joint jogging refers to the movement of each specific joint of a robot, in either a positive or negative direction.

Joint 2 movement interacts with Joint 3 movement to maintain Joint 3 in the same orientation.

In order to assist operators, some robots have positive and negative directional markings located directly on the mechanical unit for reference



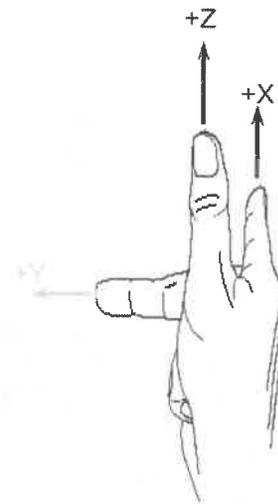
6: Jogging

6.2.4 Right Hand Rule and Cartesian Coordinate Systems

In mathematics and physics, the right-hand rule is a common mnemonic to aid in understanding three-dimensional vector relationships. In robotics, with proper positioning of the hand, the right-hand rule can also be used in remembering which direction each frame vector should be pointing.

FANUC's use of the right-hand rule assigns the X-plane to the index finger, the Y-plane to the middle finger, and the Z-plane to the thumb as shown to the right.

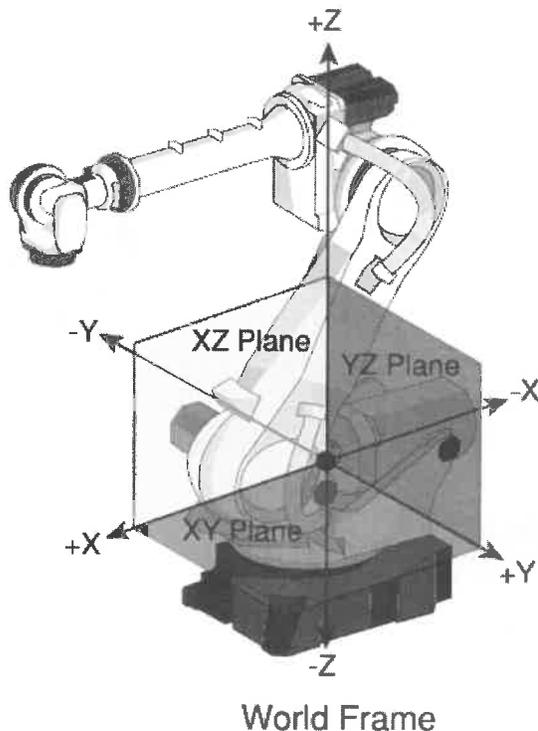
If the right hand is positioned with the index finger pointing toward the front of the robot (usually opposite of the base cables) the X, Y, and Z of the World frame can easily be recalled.



6.2.5 Jogging in World Frame

The world frame is a default frame of the robot and cannot be changed. It is the basis for all taught positions and all taught frames within the robot. On most FANUC serial-link robots, the origin of World frame is typically designated as the intersection of the center of the robot base with a plane level to the Joint 2 motor.

This origin, along with the X, Y, and Z directions, is illustrated in the image below. Robot positions that reference the World frame reflect the position of the robot's Tool Center Point (TCP) in X, Y, Z, W, P, and R relative to the origin of this frame.

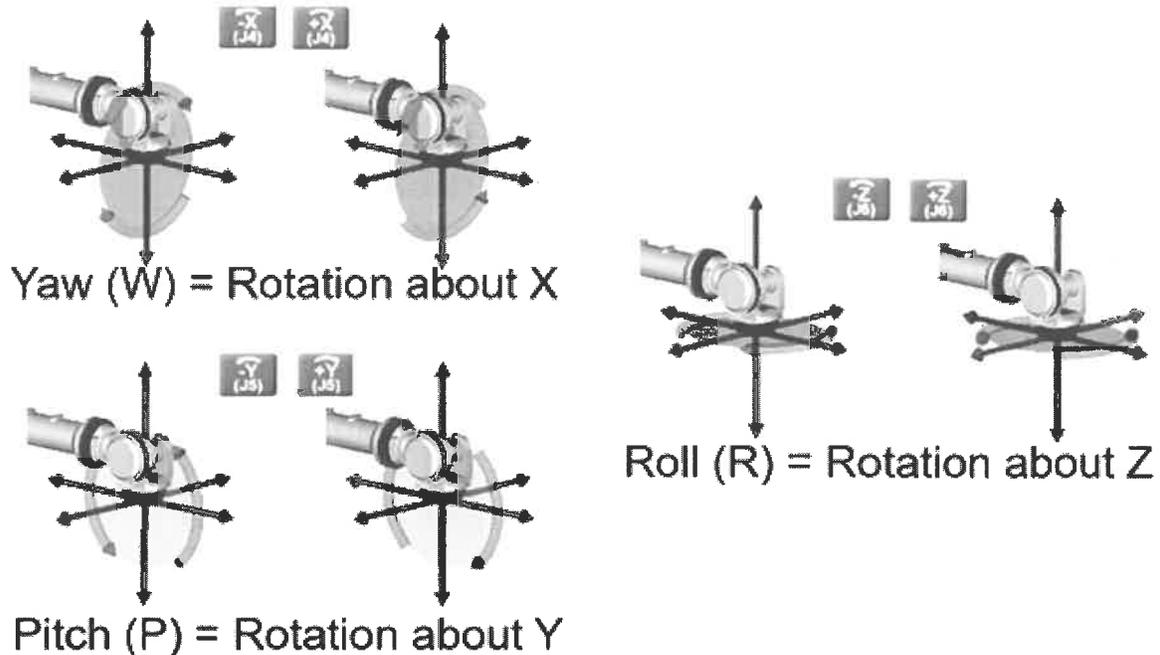


Positive and Negative references dictate the direction of travel (robots perspective):

- X+ moves the end of arm straight forward
- X- moves the end of arm straight back
- Y+ moves the end of arm laterally to the left
- Y- moves the end of arm laterally to the right
- Z+ moves the end of arm straight up
- Z- moves the robot straight down

While jogging in WORLD the motion is linear. The robot will coordinate the movement of each joint to maintain a straight line motion at the end-of-arm tooling. This movement is based on the World Coordinate System orientation which is squared to the robot.

Besides straight in-line movements you can also jog about a coordinate axis. You can rotate the end of arm about X, Y, and Z.



6.3 Jogging the Robot

Prior to jogging the robot ensure the following conditions are satisfied:

- All personnel and unnecessary equipment are out of the workcell.
- All EMERGENCY STOP faults have been cleared.
- All other faults have been cleared and the fault light is not illuminated.
- The MODE SELECT switch is in the T1 or T2 position.



WARNING

Make certain that all safety requirements for your workplace have been followed; otherwise, you could injure personnel or damage equipment.

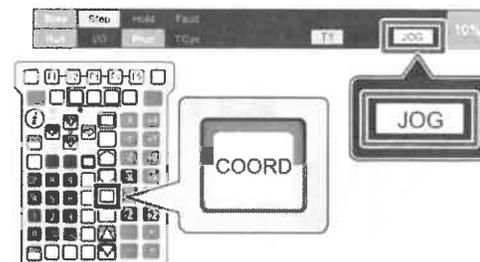
Procedure 6-1 Jogging the Robot

- 1 Turn the teach pendant ON/OFF switch to the ON position

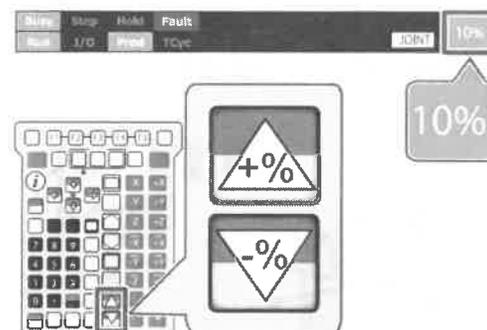


6: Jogging

- 2 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



- 3 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,

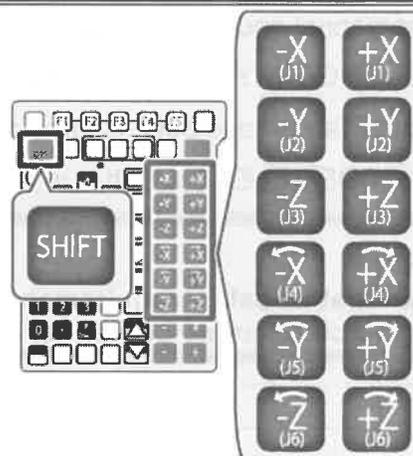


- 4 Hold the iPendant and continuously press and hold the DEADMAN switch on the back in center position.

⚠ 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.

- 5 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.



- 6 When you are finished jogging, turn the teach pendant ON/OFF switch to OFF, and release the DEADMAN switch.

6.4 Positional Data

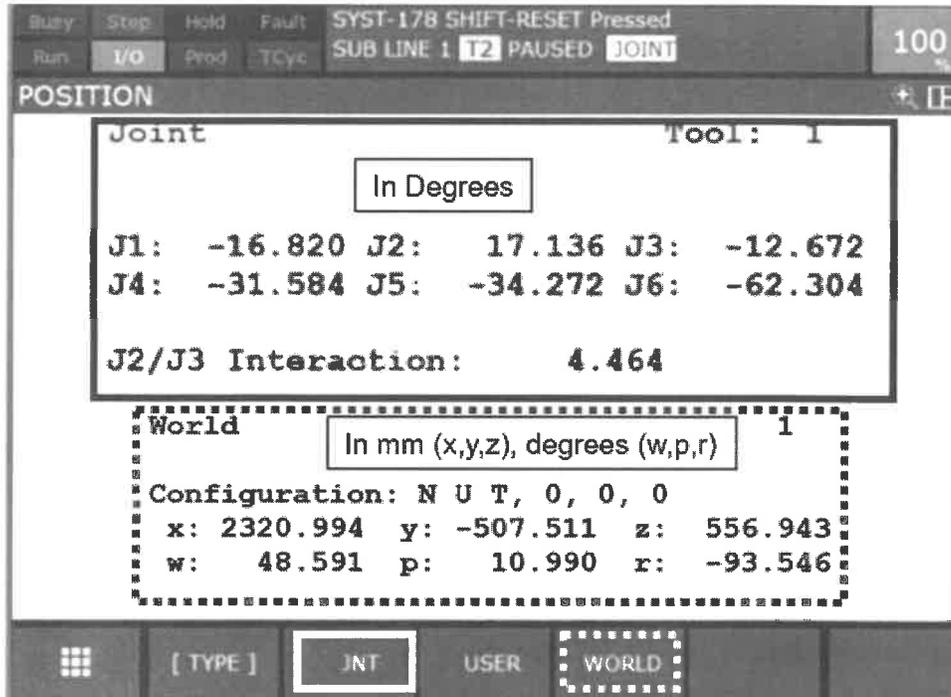
The POSITION screen displays positional information in joint angles or Cartesian coordinates. The positional information on this screen is updated continuously when the robot moves. The information displayed on these screens is for display only; you cannot change it.

6: Jogging

Handling Tool

The positional data displayed below shows joint representation in the upper box and the same position in Cartesian in the lower box.

To display the Position screen depress the POSN key at the bottom of the iPendant or access through the Menus (Menu, -0- Next, 4D Graphics, Position Display.)



6.5 Chapter Review

1. **What key is depressed to switch between the different jog methods?**
 - A POSN
 - B MENU
 - C FCTN
 - D COORD

2. **Assuming standard setup, if you jog the robot in World -Y which direction will the TCP move?**
 - A Straight out front
 - B Up
 - C To the robots right
 - D To the robots Left

3. **If while jogging the robot you release the shift key, or jog key what will happen?**
 - A The robot will continue to move to the position I intended.
 - B The robot will execute a controlled stop.
 - C The robot will execute an EMERGENCY STOP.
 - D These keys are not required to jog the robot.

Lab 3

Jogging in Joint and World

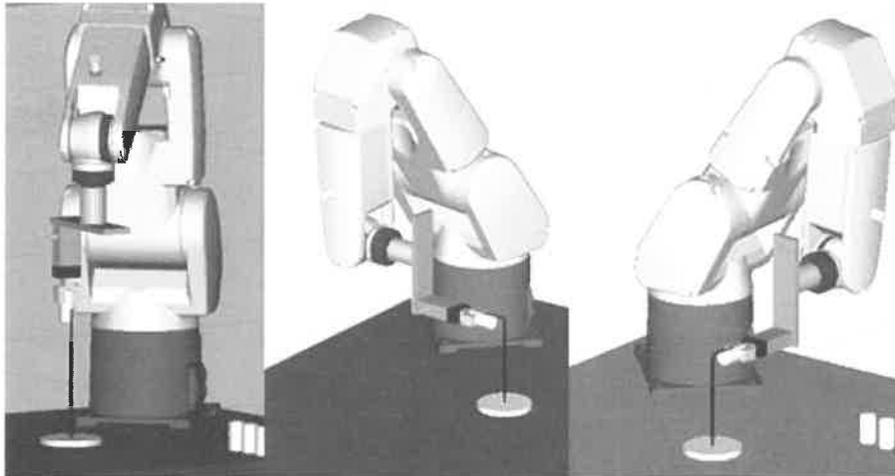
Student Name: _____

Assignment: The student will:

- Jog the robot to a predetermined point in Joint
 - Jog the robot to a predetermined point in World
-

Condition: Given a FANUC Robot and a controller

- Step:**
- 1 Utilizing the reference pointer in the EOAT and the reference pointer placed in the cell by your instructor, jog the tip of the EOAT reference pointed to the tip of the pointer in the cell utilizing JOINT Jogging. (See **Section 6.3 Jogging the Robot, pg. 63.**)
 - a Perform the same operation as above utilizing WORLD Jogging and approach from a different angle.
 - 2 Repeat the lab until you have jogged the robot to reflect each of the postures shown below.



Completed:

Instructor: _____

7 FILE MANIPULATION

7.1 Learning Objectives

- Controller Memory Overview
- Understand the Types of Backups
- Perform a File Backup and Restore
- Perform an Image Backup and Restore

7.2 Controller Memory Overview

There are three kinds of controller memory:

- Dynamic Random Access Memory (DRAM)
- A limited amount of battery-backed static/random access memory (SRAM)
- Flash Programmable Read Only Memory (F-ROM)

DRAM	<ul style="list-style-type: none"> • DRAM memory is not battery backed up and volatile. <ul style="list-style-type: none"> • Memory contents do not retain their stored values when power is removed. • DRAM memory is also referred to as temporary memory (TEMP). • The system software (normal controller operations) is executed in DRAM memory.
SRAM	<ul style="list-style-type: none"> • SRAM memory is battery backed up and nonvolatile as long as the batteries are working. <ul style="list-style-type: none"> • Memory contents retain their stored values when power is removed. • Also referred to as CMOS or as permanent memory (PERM). • SRAM Memory has two parts: <ul style="list-style-type: none"> • The TPP memory pool, which contains the teach pendant programs. • The permanent (PERM) memory pool, which contains system variables and KAREL variables. • A portion of SRAM memory can be defined as a user storage device called RAM Disk (RD:).
<p> CAUTION</p> <p>Data in SRAM can be lost if the battery is removed or loses its charge, or if new core software is loaded on the controller. The SRAM memory will last for 30 minutes without the battery when power is off. To prevent loss of data, back up or copy all files for permanent storage.</p>	
F-ROM	<ul style="list-style-type: none"> • Flash ROM (F-ROM or FROM disk) is not battery-backed but is non-volatile

7: File Manipulation

	<ul style="list-style-type: none"> • All memory contents in F-ROM retain their stored values when power is removed. • F-ROM is used for permanent storage of the system software. • F-ROM is file oriented and all of it is allocated to the flash file system. • The flash file system contains both system and user files. • F-ROM is also available for user storage as the FROM device (FR:).
--	--

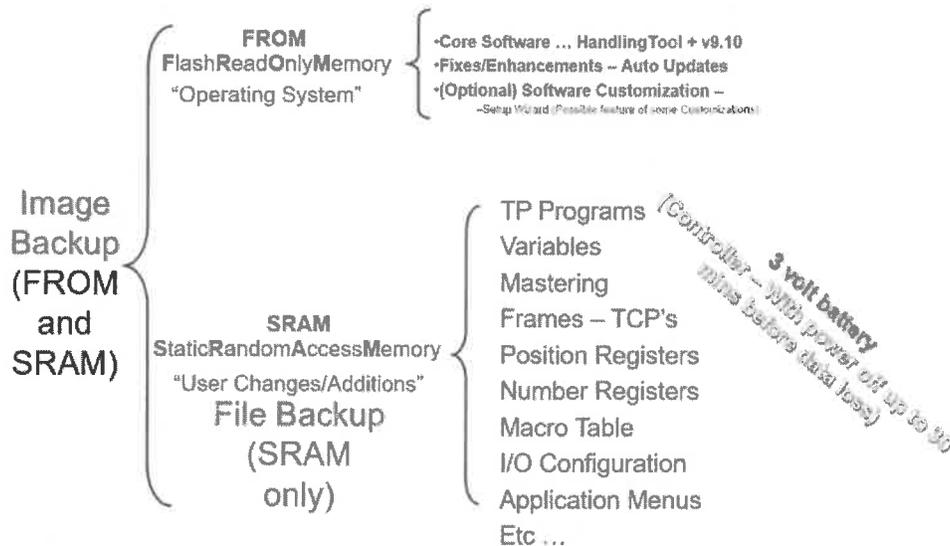
7.3 Types of Backups

File Backup (SRAM):

- Contains all user additions and changes unique to that robot:
 - Frames
 - All Register Data
 - Menu Settings
 - Macro Table
 - Variables
 - Position Registers
 - I/O Configurations
 - TP Programs
- A user can select individual items for restoration if required.

Image Backup (F-ROM and SRAM):

- Contains all of the above items plus the Operating System, Maintenance Updates, and Customization software of the robot.
- An image is an exact copy of the robot memory.
 - You cannot pick individual files to restore, it's all or nothing



7.4 Preparing to Create a Backup

Performing a File and/or an Image Backup is a fairly simple process, however there are a few important items to consider prior to saving your information. These items are:

- Setting the Default Device.
- Formatting the device.
- Creating a directory structure
- Performing the backup.

NOTE: It is not always necessary to perform each of the above depending on the specific circumstances of the backup.

7.4.1 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 memory card.

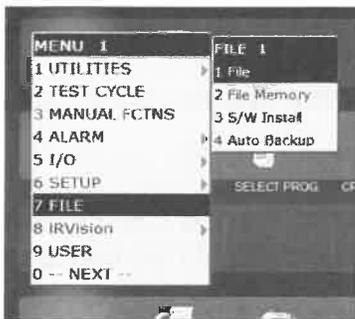
You can set the default device to:

- Memory Card (MC:) (not available on the R-30iB Mate Plus Controller)
- USB Memory Stick (UD1: / UT:1)
- Flash File Storage disk (FR:) - built-in storage for controller system files.
- Memory device (MD:) - treats the controller's program memory as if it were a file device.
- MF device (MF:) - a composite device that searches the RAM Disk (RD:) and flash file storage disk (FR:) devices, in that order, for a specified file.

NOTE: Once you set the default device it remain the active device until changed.

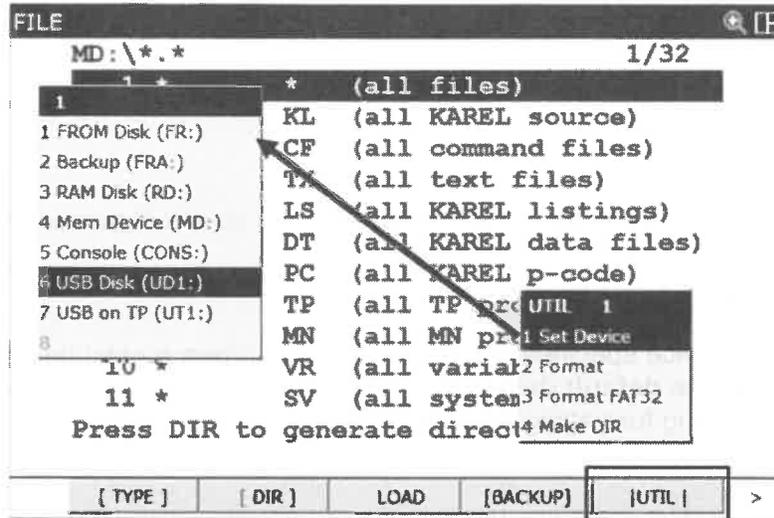
Procedure 7-1 Setting the Default Device

- 1 Press Menu.
- 2 Cursor down to File.
 - a Press the right arrow.
 - b Select File from the submenu.

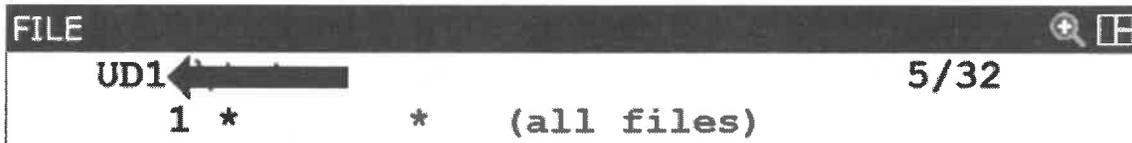


7: File Manipulation

- 3 Press F5 [UTIL].
- 4 Select "Set Device"
- 5 Move the cursor to the required device and press Enter.



- 6 The selected device will display along the top line of the screen.



7.4.2 Formatting the Device

You must format the FLASH ATA memory card or USB memory stick **only before you use them for the first time**. Formatting the Device after the first use will result in losing all of the data stored on the device. The device may be formatted at FAT or FAT 32.

Format the Device as FAT if:

- The device is less than 2Gbytes
- More than 255 files do not need to be stored in the root directory of the media.
 - Larger quantities of files can still be stored in sub-directories
- It is necessary to write files to the media quickly.

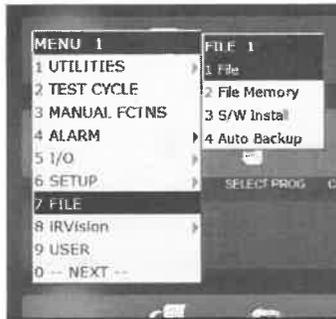
Format the Device as FAT32 if:

- There is no limit to the number of files stored in the root directory of the media.
- If the device is larger than 2.0 GB.

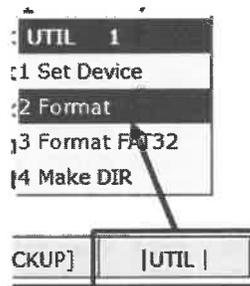
NOTE: USB Memory sticks cannot be formatted when plugged into the USB connector on the iPendant (UT1:) They must be formatted on a PC or when plugged into the front panel USB connector on the controller.

Procedure 7-2 Formatting the Device

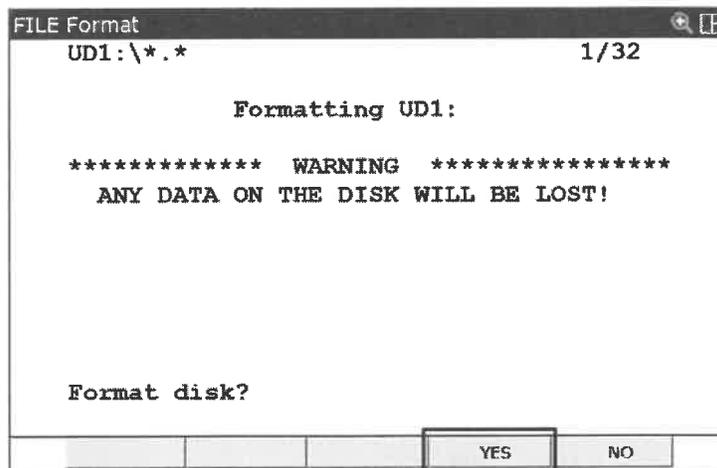
- 1 Press Menu.
- 2 Cursor down to File.
 - a Press the right arrow.
 - b Select File from the submenu



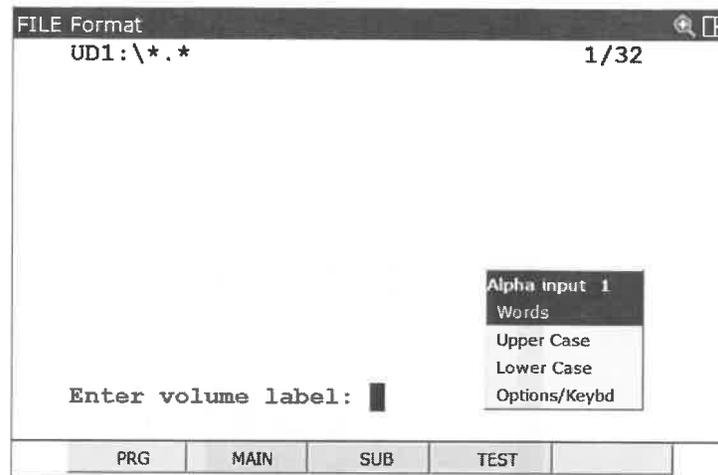
- 3 Press F5 [UTIL] and select Format (FAT) or Format FAT32 to format the device.



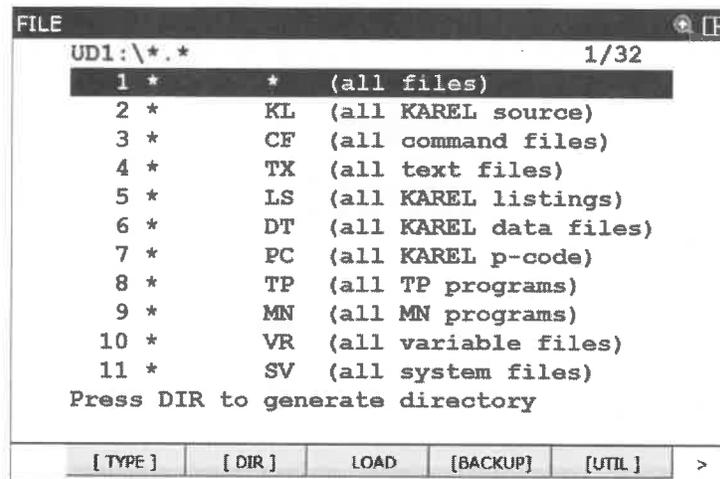
- 4 Choose F4 [YES] to complete the format.



- 5 Enter a name for the storage device or press enter to finish the format.



- 6 The screen will return to the File screen once complete.

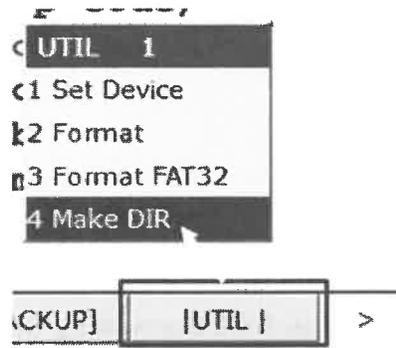


7.4.3 Subdirectories

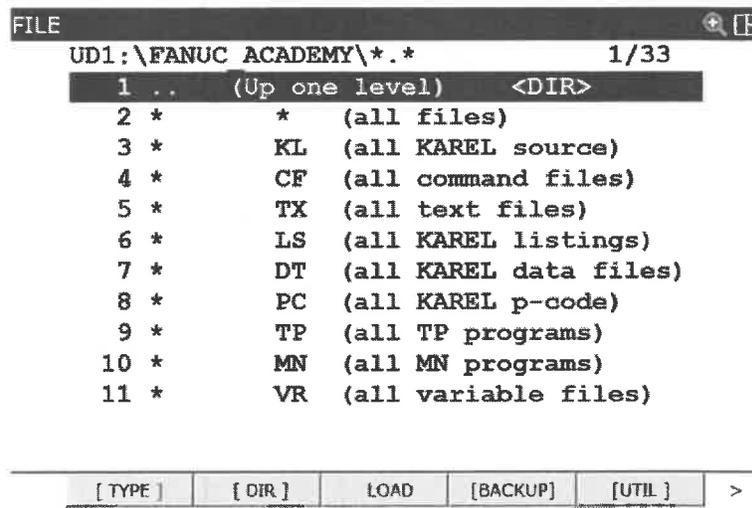
Subdirectories can be used to organize programs or files on the controller. Subdirectories can be created on the FILE menu.

Procedure 7-3 Creating a Subdirectory

- 1 Press Menu.
- 2 Cursor down to File.
 - a Press the right arrow.
 - b Select File from the submenu
- 3 Press F5 [UTIL] and select Make DIR



- 4 After naming the directory, press Enter. You will automatically be in the new directory that you created.



- 5 To exit a subdirectory highlight "Up one level" and press enter, left arrow, or right arrow.
- 6 To enter a subdirectory highlight the subdirectory name and press enter, left arrow, or right arrow.

7.5 File Backup

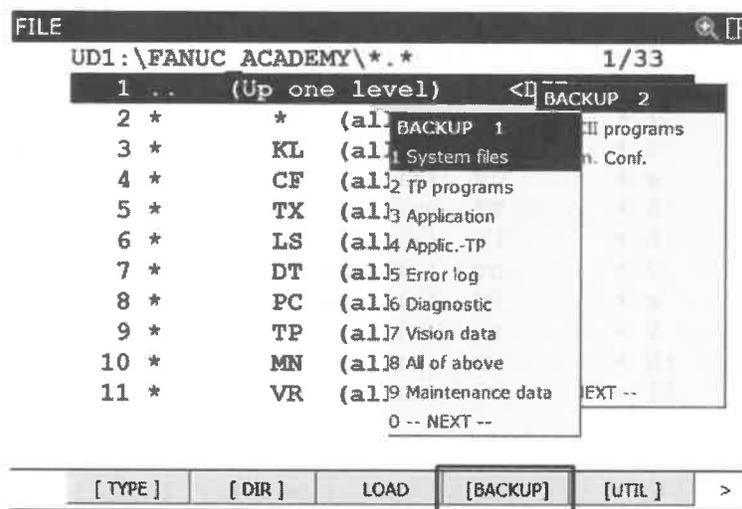
When you back up a file, you save it from controllers SRAM memory to the default device (chosen storage media) so that you have a second copy of the file. When performing a File Backup you are specifically saving program, system, application, diagnostic, and error log files to the default device using the FILE screen. Application software (such as HandlingTool, SpotTool, PaintTool, etc.) will **NOT** be backed up when using the FILE menu.

Remember from the prior sections you must set the default device and the device should only be formatted the first time it is used. Also, it is not absolutely necessary to add subdirectories, however they are a good tool to organize your backup data.

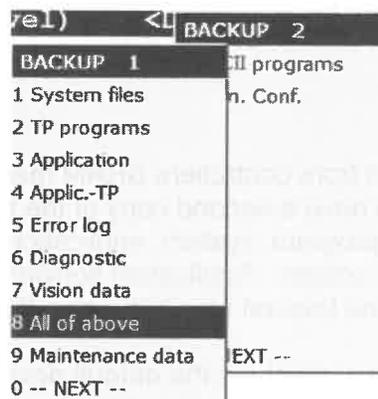
NOTE: This section covers how to create a complete File Backup, individual File Backup will be covered in later Chapters.

Procedure 7-4 Creating a File Backup

- 1 Press Menu.
- 2 Cursor down to File.
 - a Press the right arrow.
 - b Select File from the submenu
- 3 Set the default device, see 7.4.1 **Setting the Default Device**.
- 4 Format the device (if necessary), see 7.4.2 **Formatting the Device**.
- 5 Create subdirectories (if necessary), see 7.4.3 **Subdirectories**.
- 6 Press F4 [BACKUP].

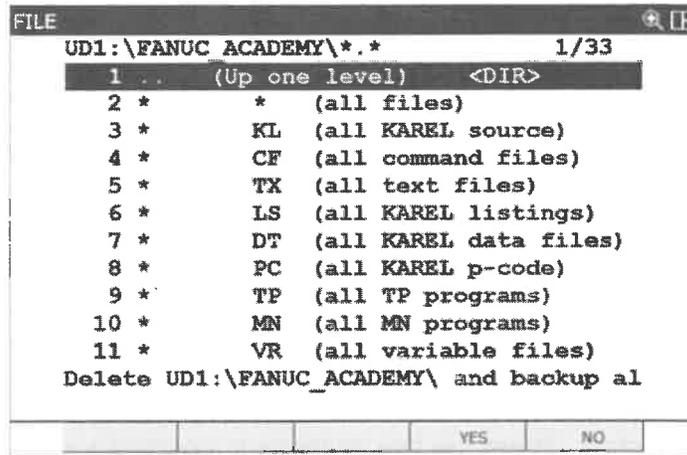


- 7 Highlight All of Above and press enter, or Press the number 8 on the keypad.

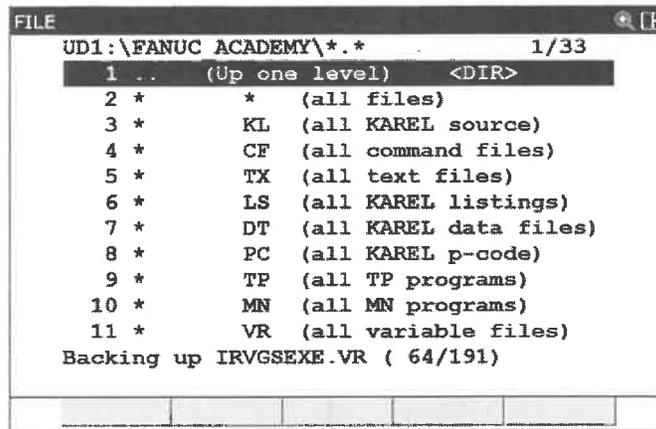


- 8 Choose F4 [YES] to delete UD1.
 - a By selecting YES you are deleting the contents of the directory you have navigated too. This is necessary because the same File names are saved with each backup. The same file names cannot exist in the same directory.

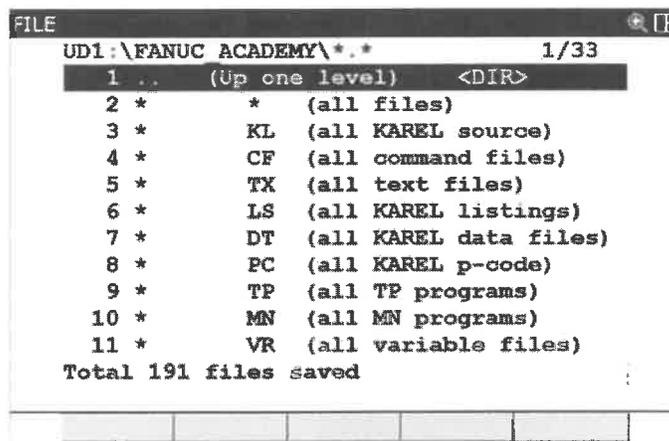
7: File Manipulation



9 The backup will start, you will see a scrolling list of what is being saving.



10 Once complete you see a total number of files saved, similar to below.



7: File Manipulation

7.6 File Restore

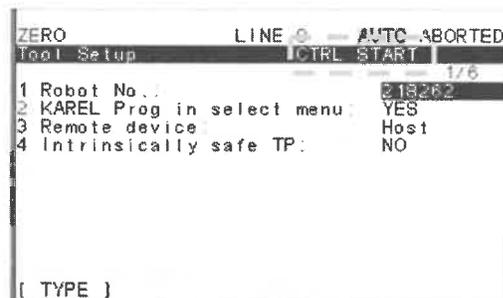
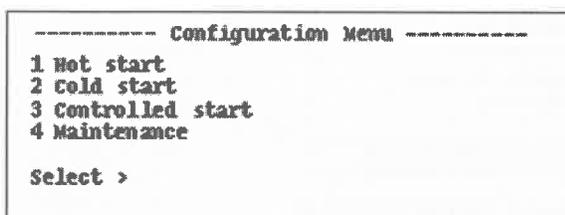
You restore files from a disk when you have previously backed up the files using BACKUP on the FILE screen. A File Restore overwrites the existing data on the controller.

It is very common that not every file contained in the backup is restored. For example you may see after the restore 121/135 files restored. The files that were not restored will generally be System Macros and KCL files that are write protected and cannot be overwritten.

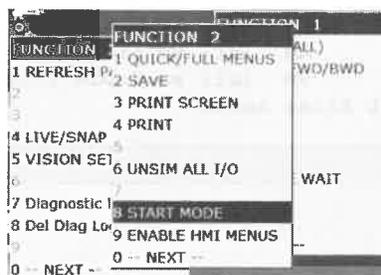
NOTE: This section covers how to restore a complete File Backup, individual file restoration will be covered in later chapters.

Procedure 7-5 Restoring a File Backup

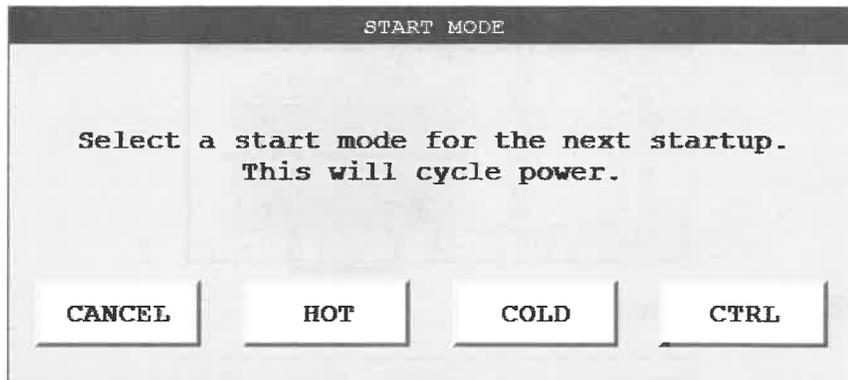
- 1 Perform a Controlled start. There are two methods to conduct a controlled start.
 - a If your system is powered off proceed to **Step 2**.
 - b If you system is powered up proceed to **Step 3**.
- 2 Perform a Controlled Start with system powered off.
 - a Press and hold, the PREV and NEXT keys on the teach pendant.
 - b Turn on the controller.
 - c At the configuration menu select Controlled Start, you will see a screen similar to below.



- 3 Perform a Controlled Start with system powered on.
 - a Press the Function key
 - b Press -0- for next
 - c Select -8- Cycle Power or Start Mode



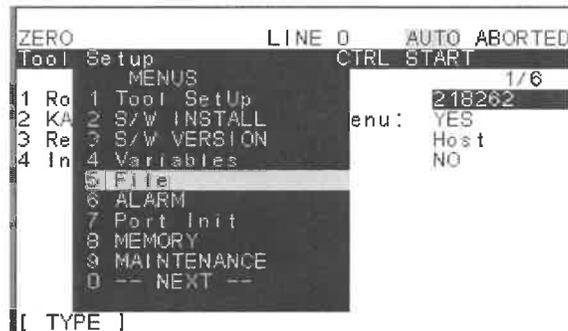
- d On the menu select CTRL (Controlled Start).



- e If working on a robot equipped with LR HandlingTool Software you will have to manually turn the controller off and back on. The controller will boot up in the controlled start mode.
- f If working on a robot equipped with standard HandlingTool software the controller will shut down and boot up in controlled start mode automatically.
- g If you do not see the CTRL option on the menu, your software is not equipped with this feature. Perform Step 2 to access the Controlled Start mode.

4 Press Menu.

5 Cursor down to File and press Enter.



6 Set the default device, see **7.4.1 Setting the Default Device**

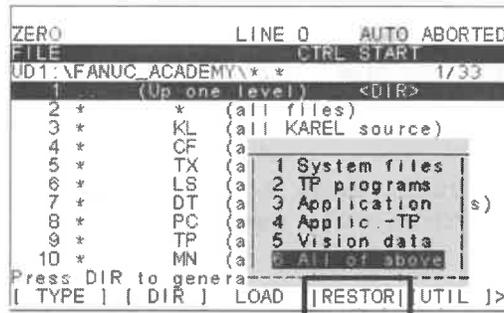
7 Navigate to the appropriate directory, see **7.4.3 Subdirectories**

8 Press F4, [RESTOR].

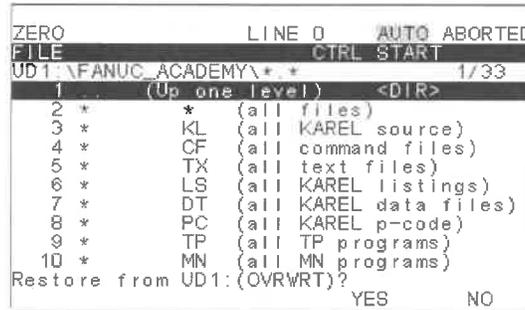
- a If F4, [BACKUP] is displayed, press FCTN and then select RESTORE/BACKUP to display F4, [RESTOR].

9 Highlight All of above and press Enter.

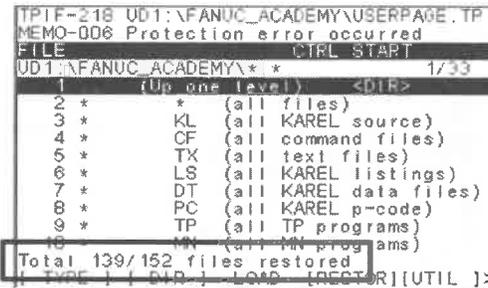
7: File Manipulation



10 Press F4 [YES] to begin the restore.



11 Once complete, you will see a number of files that have been restored.



12 To operate the robot, cycle controller power:

- a Press FCTN.
- b Select Start (Cold).
- c Press ENTER.



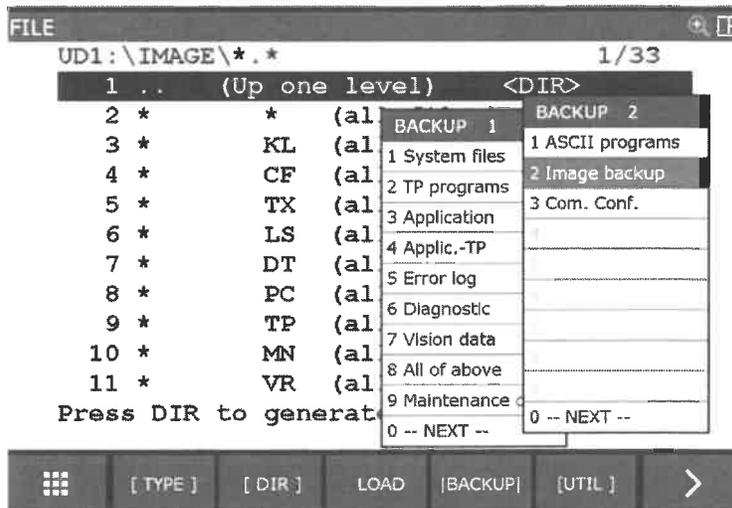
7.7 Image Backup

The Image Backup feature allows you to make an image of the F-ROM and SRAM controller memory modules. The image is stored in a number of files (*.img) on a selected destination device.

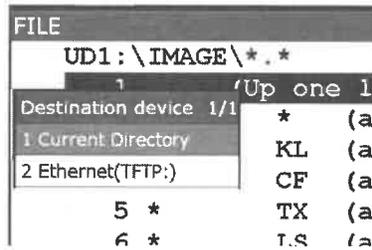
Image Backup is available from the FILE menu when the system is in Cold start or Controlled start mode. After Image Backup is selected from the menu, the actual backup occurs the next time the controller is powered up.

Procedure 7-6 Creating an Image Backup

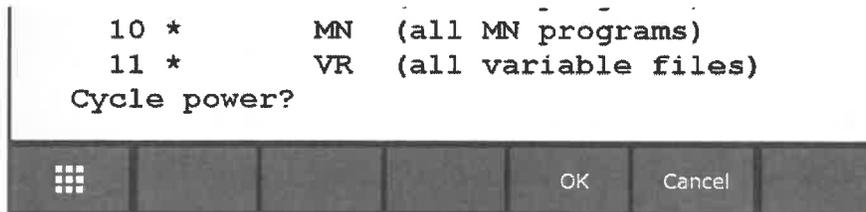
- 1 Press Menu.
- 2 Cursor down to File.
 - a Press the right arrow.
 - b Select File from the submenu
- 3 Set the default device, see 7.4.1 **Setting the Default Device**.
- 4 Format the device (if necessary), see 7.4.2 **Formatting the Device**.
- 5 Create subdirectories (if necessary), see 7.4.3 **Subdirectories**.
- 6 Press F4 [BACKUP].
- 7 On the second page on choices select Image Backup,
 - a Either highlight the selection and press Enter or depress the number 2 key on the keypad.



- 8 If you are navigated to the correct subdirectory, press enter to select current directory. See the file path directly above the prompt to view the directory path.



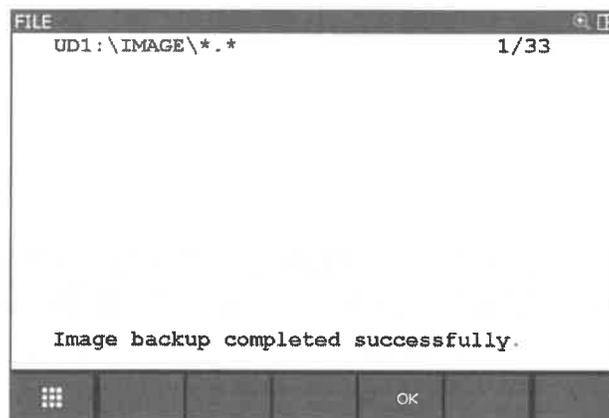
- 9 Press F4 [OK] to complete the power cycle
 - a If working on a robot equipped with LR HandlingTool Software you will have to manually turn the controller off and back on. On power up, the image backup starts automatically.
 - b If working on a robot equipped with standard HandlingTool software the controller will shut down and boot up automatically. On power up, the image backup starts automatically.



- 10 While the image files are written you will see a screen similar to below.

```
WRITING MC:\FROM00.IMG  
WRITING MC:\FROM01.IMG  
.....  
WRITING MC:\SPAM01.IMG  
DONE!
```

- 11 Once the Image has been successfully written the controller will cycle power again and you will see screen similar to below. Press F4 [OK] to resume normal operations.



7.8 Image Restore

The Image Restore feature allows you to load the saved F-ROM and SRAM controller memory modules back into the controller. The image is stored in a number of files (*.img) on a selected destination device.

NOTE: Stored within the image is the hardware serial number of the controller that was used to produce the image. If this image is restored on a different controller, an OPTN-027 alarm (ID Mismatch) will be posted on every COLD start. To stop OPTN-027 alarms from being posted, the ID of the image must be changed to match the new ID of the controller. Contact FANUC America Corporation for full details.

Procedure 7-7 Restoring an Image

- 1 Press F1 and F5 while turning on the power disconnect circuit breaker.
- 2 You will see a screen similar to below.
 - a Select Controller backup/restore and press ENTER.

```

*** BOOT MONITOR ***      UD
Base version V8 30P /13 [Release 3]
***** BMON MENU *****
1. Configuration menu
2. All software installation(MC)
3. All software installation(Ethernet)
4. Controller backup/restore
5. All software installation(USB)
6. Maintenance
7. All software installation(Ethernet)
8. All software installation(USB)

Select :

```

- 3 Select restore controller as Images.

```

*** BOOT MONITOR ***      UD
Base version V8 30P /13 [Release 3]

Select : 4

***** BACKUP RESTORE MENU *****
0 Return to MAIN menu
1 Emergency Backup
2 Backup Controller Images
3 Restore Controller Images
4 Download to CF card menu

Select : 3

```

- 4 Select the appropriate default device from the list displayed.

```

*** BOOT MONITOR ***      UD
Base version V8 30P /13 [Release 3]
4 Bootstrap to CFG MENU

Select : 3

** Device selection menu ****
1 Memory card(MC)
2 Ethernet(TFTP)
3 USB(UD1)
4 USB(UT1)

Select :

```

7: File Manipulation

- 5 You will be prompted to select the appropriate subdirectory if applicable.
 - a Choose the directory path from the list.
 - b Once the path shown is properly setup, Press 1 to select current directory.
- 6 The controller will verify that the target directory F-ROM and SRAM correspond with controller memory size.

CAUTION

The following operation erases all of SRAM and FROM. Be sure you are prepared to erase all of SRAM and FROM before you execute.

- a Press 1 for YES when you are ready to begin the restore
- 7 Restoring will take several minutes (approximately 3–6 minutes). You will see messages displayed on the screen stating that SRAM and FROM are being cleared and that files are being loaded. You will see a screen similar to the following.

```
*** BOOT MONITOR ***          UD
Base version V8.30P/13 [Release 3]
Reading FROM59.IMG ... Done
Reading FROM60.IMG ... Done
Reading FROM61.IMG ... Done
Reading FROM62.IMG ... Done
Clearing SRAM (2M) ...
done
Reading SRAM00.IMG ... Done
Reading SRAM01.IMG ... Done
-- Restore complete --

Press ENTER to return >
```

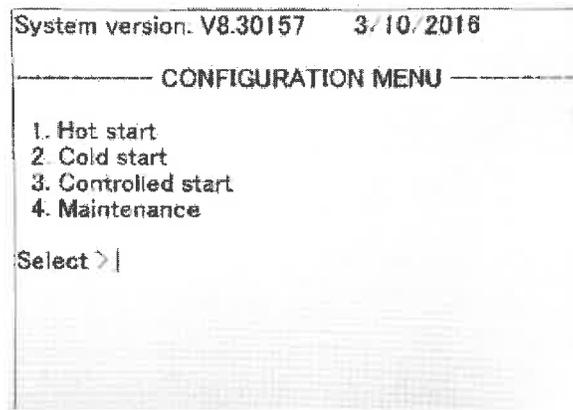
- 8 Once the load has completed Press Enter to return to the BMON screen.
- 9 From the BMON screen select Configuration Menu.

 **NOTE:** DO NOT choose INIT Start, this will cause all memory to be cleared from the controller.

```
*** BOOT MONITOR ***          UD
Base version V8.30P/13 [Release 3]
BMON MENU
1. Configuration menu
2. All software installation(ISO)
3. INIT start
4. Controller backup/ restore
5. Hardware diagnosis
6. Maintenance
7. All software installation(Ethernet)
8. All software installation(USB)

Select :
```

- 10 From the Configuration Menu Screen select Cold Start and resume normal operations.

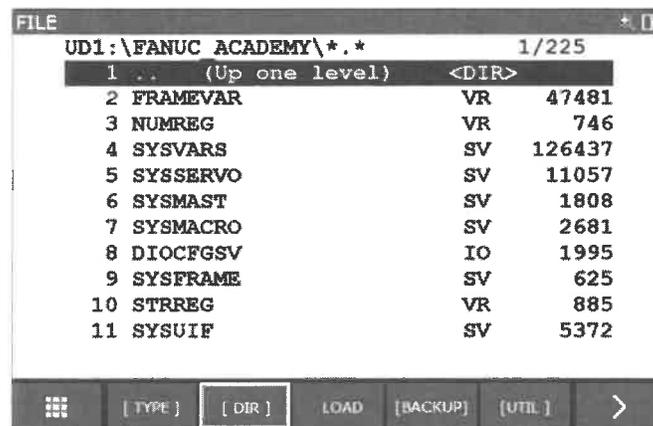


7.9 Generating a Directory

A directory is a list of files on a specific storage device. You can display a directory of files on the default device. Some devices contain hundreds of files, you can display all files, or a subset of files. When you generate a directory of files, you can choose from among the available file types by pressing F2 [DIR] or selecting one of the file types listed on the File Screen.

Procedure 7-8 Generating a Directory of Files

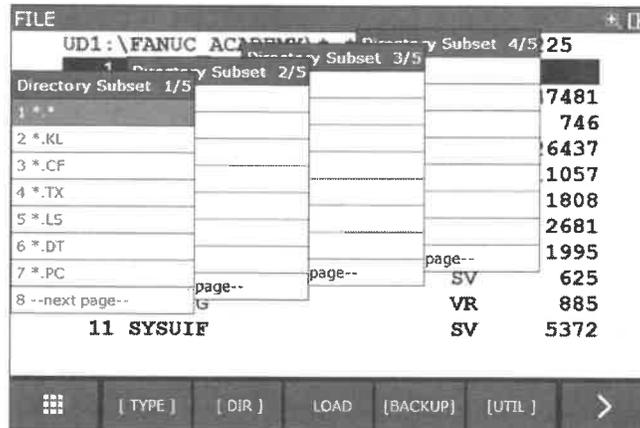
- 1 Press Menu.
- 2 Cursor down to File.
 - a Press the right arrow.
 - b Select File from the submenu
- 3 Set the default device, see 7.4.1 **Setting the Default Device**.
- 4 Navigate to the appropriate subdirectory (if necessary), see 7.4.3 **Subdirectories**.
- 5 Press F2 [DIR].



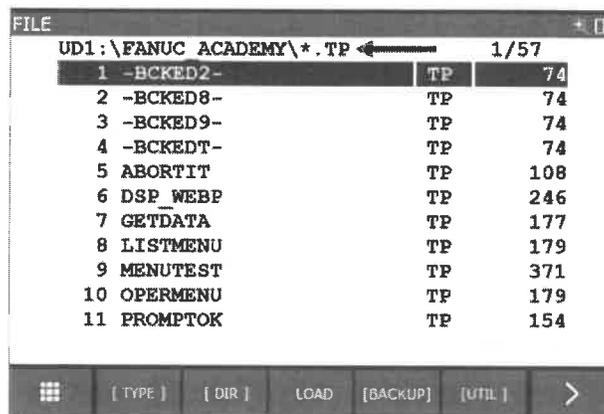
- 6 On the popup menu select the required file extension.

7: File Manipulation

- a Selecting *.* will display all files in the current directory
- b Selecting a specific extension will display only the selected file types.



- 7 The selected extension contained within the set directory will be displayed.



7.10 Chapter Review

1. What must be done prior to executing any type of backup or restore?

- A Set the Default Device/Verify the correct device is set
- B Make a Directory
- C Nothing
- D Format the Device

2. What key sequence is depressed and held to access the Controlled Start Screen while powering on the controller?

_____ and _____

3. What key sequence is depressed and held to access the BMON screen while powering on the controller?

_____ and _____

4. What type of memory is saved when performing an Image Backup?

- A F-ROM Only
- B SRAM Only
- C F-ROM and SRAM
- D DRAM

5. What type of memory is saved when performing a File Backup?

- A F-ROM Only
- B SRAM Only
- C F-ROM and SRAM
- D DRAM

6. When restoring a File Backup, SRAM files on the controller are overwritten.

- A True
- B False

7. When restoring an Image Backup, F-ROM and SRAM files on the controller are erased.

- A True
- B False

7: File Manipulation

HandlingTool

Lab 4

File Backup / Restore

Student Name: _____

Assignment: The student will:

- Create a File Backup
 - Restore a File Backup
-

Condition: Given a FANUC Robot and a controller

- Step 1** Place the robot at the ZERO position.
- 2** Create a File Backup. (**See Section 7.5 File Backup, pg. 75**)
- a** You must create a directory and name it "File_BU_(Date Taken)".
Ex: File_BU_061318
- 3** Restore the File Backup. (**See Section 7.6 File Restore, pg. 78**).

Completed:

Instructor: _____

Lab 5 Image Backup

Student Name: _____

Assignment: The student will:

- Create an Image Backup
-

Condition: Given a FANUC Robot and a controller

- Step 1** Place the robot at the ZERO position.
- 2** Create an Image Backup. (**See Section 7.7 Image Backup, pg. 81**).
- a** You must create a directory and name it "Initial_Image_(Date Taken)". Ex: Initial_Image_061318
- 3** Save this image as you will be loading it at the end of the class.

Completed:

Instructor: _____

8 INITIAL SETUP

8.1 Learning Objectives

- Setup Robot Axis Limits
- Understand Payload
- Verify Mastering

8.2 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, and/or cable and hose lengths

There are three methods used to prevent the robot from going beyond the necessary motion range:

Axis limit 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 or hard stops are contacted two to three degrees beyond the software limits. Overtravel switches for axis 1 are available as an option.
Axis limit switches (optional)	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 for axis 1 are available as an option.
Axis limit hard stops	Axis limit hard stops are physical barriers that are located two or three degrees beyond the overtravel limit switch or software setting on the three major axes. The robot cannot move beyond a hard stop. 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.

WARNING

Do not use axis software limits as the only method for restricting robot motion. Change the hard stops to match the software modifications; otherwise, you could injure personnel or damage equipment.

CAUTION

Changing the axis limits will affect the robot work area, and could change robot motion. Anticipate the effects of changing axis limits before you change them; otherwise unexpected results could occur, such as errors in previously recorded positions.

8: Initial Setup

8.3 Axis Limits Software Settings

As mentioned above, one of the ways to limit the area the robot can travel is axis limit software settings. When manipulating these values an upper and a lower limit are defined.

Procedure 8-1 Setting Axis Limit Software Settings

- 1 Press MENU
- 2 Select -0- Next
- 3 Cursor to System.
 - a On the fly out menu cursor down to Axis Limits, you will see a screen similar to below.

SYSTEM Axis Limits				1/84
AXIS	GROUP	LOWER	UPPER	
1	1	-185.00	185.00 dg	
2	1	-60.00	76.00 dg	
3	1	-132.04	180.00 dg	
4	1	-360.00	360.00 dg	
5	1	-125.00	125.00 dg	
6	1	-360.00	360.00 dg	
7	0	0.00	0.00 mm	
8	0	0.00	0.00 mm	
9	0	0.00	0.00 mm	
10	0	0.00	0.00 mm	

Item	Description
AXIS	This item displays the number for each axis
GROUP	This item indicates to which group an axis belongs. A "0" indicates the robot does not have these axes.
LOWER	This item displays the lower limits of each axis, or the axis limits, in a more negative direction. You can change these values.
UPPER	This item displays the upper limits of each axis, or the axis limits, in a more positive direction. You can change these values.

- 4 Move the cursor to the axis limit to set.
 - a Type the new value, press enter
 - b Repeat for other settings.

NOTE: Positive values are assumed, negative values must be specified.

NOTE: The lower value is not always negative, and the upper value is not always positive. However, ensure the smaller number is in the LOWER column and the larger number is in the UPPER column.

- 5 Turn off the controller and then turn it back on again so the new information can be used.

8: Initial Setup

HandlingTool

8.3.1 Axis Limits Software Settings Faults

Once the limits have been established the robot will not move beyond the limits. If the robot hits the predetermined limit it will perform a controlled stop and a message (similar to the right) will be displayed on the Alarm Status Line of the iPendant. Within the message, the affected motion group (G:[n]) and axis/axes (A:[n]) is displayed.

JOG -013 Stroke limit (G:1 A:6)
 JOG -013 Stroke limit (G:1 A:5)
 JOG -013 Stroke limit (G:1 A:4)
 MOTN-017 Limit error (G:1, A:2,3)
 JOG -013 Stroke limit (G:1 A:2)
 JOG -013 Stroke limit (G:1 A:1)

To recover from hitting an axis limit
 simply jog the limited axis in the opposite direction of the limit.

8.4 Payload

It is **vitaly important** that the payload data used by the robot be as accurate as possible. When setting payload schedules a good thought to keep in mind is, "perfect will be just fine."

Payload tells the robot:

- What it is moving (mass)
- Where is the mass centered (COG) relative to the faceplate
- How is that load going to affect the robots motions (inertia).

Item	Units	Description	
Mass	kg	This item is the mass of the end-of-arm tooling.	
COG X	cm	This item is the up and down offset of the payload center of gravity from the center of the faceplate. Positive (+) values are up.	
COG Y	cm	This item is 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.	
COG Z	cm	This item is the offset of the payload center of gravity from the center of the faceplate. Positive (+) values are out from the faceplate.	
Inertia X	kgfcm ²	This item is 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.	
Inertia Y	kgfcm ²	This item is 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.	
Inertia Z	kgfcm ²	This item is 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.	

NOTE: There are 2.21 pounds in a kilogram and 0.39 inches per centimeter.

8: Initial Setup

The more accurate the values, the more effective features such as Collision Guard and Soft Float will be. Accurate values will also improve positional accuracy, cycle time, and general motion performance that are critical for today's processing.

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: If payload has not been set, the NOTIFICATIONS icon will flash in the upper right corner of the teach pendant screen. Click the flashing icon for more information.

You can define up to ten different payload schedules by default, and a maximum of 32. You can then specify a payload schedule by using the payload setup screens and by using the payload teach pendant program instructions.

8.4.1 Payload Setup

There are two ways to set up the payload schedule parameters:

- Manual entry – Values are known and entered into a schedule
- Provided engineering software or manual calculations.
- Automatic estimation using Payload Ident. (Option J669)

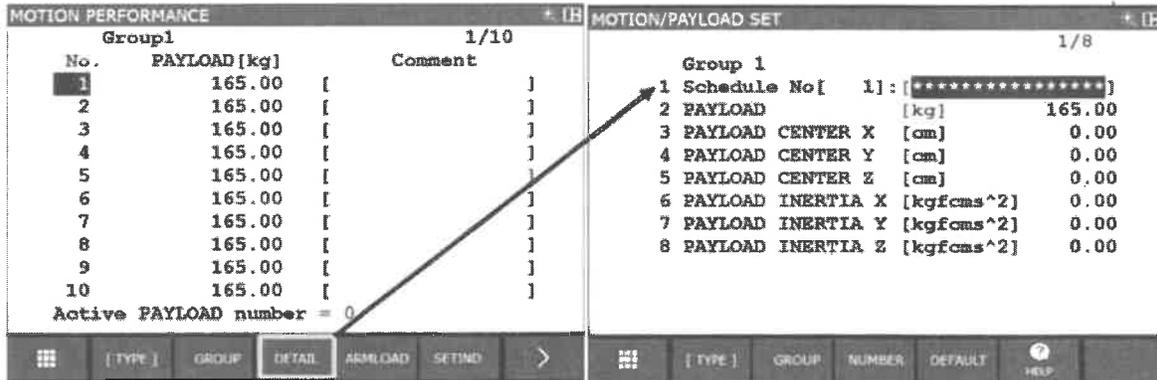
NOTE: Automatic robot payload estimation is available for some robot models. If your robot model does not have the payload setting feature, the message, "IDENT is not supported to this robot" will be displayed when you press F2, IDENT. If you change any mechanical parts on the robot, such as a motor, you need to perform payload calibration before payload estimation.

Procedure 8-2 Manually Entering Payload Data

- 1 Press Menu
 - a Press -0- NEXT
 - b Cursor down to -6- System.
 - c On the System fly out menu cursor down to MOTION and press ENTER, or press the numeric key listed next to MOTION.

MENU 2	SYSTEM 1
1 SELECT	1 Clock
2 EDIT	2 Variables
3 DATA	▶ 3 OT Release
4 STATUS	▶ 4 Axis Limits
5 4D GRAPHICS	▶ 5 Config
6 SYSTEM	▶ 6 Motion
7 USER2	7 DCS
8 BROWSER	
0 -- NEXT --	

- 2 On the payload screen cursor to the schedule to define and press F3, DETAIL.



- 3 Enter the payload data that was provided.
 - a Press F4, YES when prompted with "Path and cycle time will change, set it?"

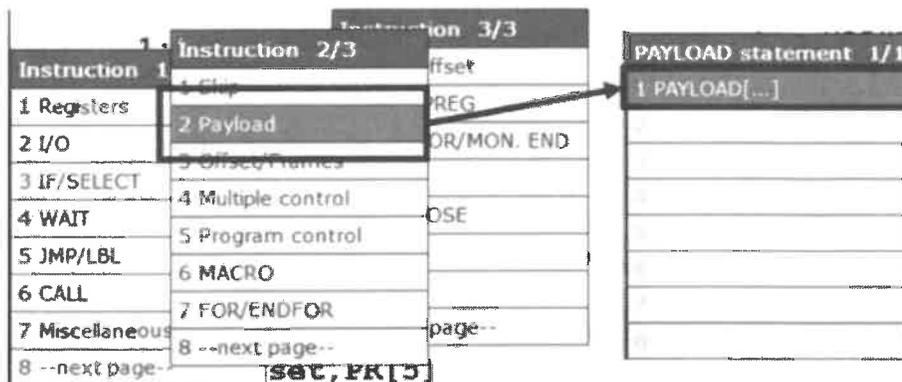
8.4.2 Payload Instructions

After Payload schedule(s) have been recorded, you can set them by pressing F5, SETIND from the payload screen and enter the schedule. Another way to set payload is with the Payload Instruction.

The Payload Instruction will set the payload data as the program executes. This is very helpful as the robot picks/drops parts or performs a tool change. As noted previously, payload data should be as accurate as possible therefore when the product the robot is moving changes so should the payload schedule.

Procedure 8-3 Programming a Payload Instruction

- 1 Place the cursor on the line in the program to add the instruction.
- 2 Press F1, [INST]
 - a If F1 does not display [INST], press NEXT.
 - b On the second page of items place your cursor on PAYLOAD.
 - c Press ENTER
 - d Highlight PAYLOAD [...], press ENTER.



8: Initial Setup

- Enter the schedule number for the Payload to set.

```

FANUC_ACADEMY
3/17
1: UFRAME_NUM=1
2: UTOOL_NUM=1
3: PAYLOAD[1]
4: R[1]=0
5: J PR[1] 100% FINE
6: LBL[1]
7: J P[1:Square] 100% CNT50
   : Offset,PR[5]
8: L P[2] 750mm/sec CNT50
   : Offset,PR[5]
9: L P[3] 750mm/sec CNT50
    
```

8.4.3 Arm Load Information

Arm Load information tells the robot what other mass is attached at J1 and J3. Think of this as wearing a backpack. As weight is added the backpack would eventually place strain on you as you walk and must be compensated for. The robot is the same, as more weight is added at the axis via valve packs, cable management systems, or other peripheral devices this will impact motion, accuracy, and functionality of certain software options.

To define Arm Load Information perform the following:

Procedure 8-4 Defining Arm Load Information

- Press Menu
 - Press -0- NEXT
 - Cursor down to -6- System.
 - On the System fly out menu cursor down to MOTION and press ENTER, or press the numeric key listed next to MOTION.

MENU 2	SYSTEM 1
1 SELECT	1 Clock
2 EDIT	2 Variables
3 DATA	3 OT Release
4 STATUS	4 Axis Limits
5 4D GRAPHICS	5 Config
6 SYSTEM	6 Motion
7 USER2	7 DCS
8 BROWSER	
0 -- NEXT --	

- On the payload screen cursor to the schedule to define and press F4, ARMLoad.

MOTION PERFORMANCE 1/10

No.	PAYLOAD[kg]	Comment
1	165.00	[]
2	165.00	[]
3	165.00	[]
4	165.00	[]
5	165.00	[]
6	165.00	[]
7	165.00	[]
8	165.00	[]
9	165.00	[]
10	165.00	[]

Active PAYLOAD number = 0

MOTION/ARMLoad SET 1/3

Group 1	[kg]
1 J2 BASE LOAD	0.00
2 J3 ARM LOAD	0.00
3 J3 CASING LOAD	0.00

- 3 Enter the ARMLOAD data that was provided.
 - a Press F4, YES when prompted with "Path and cycle time will change, set it?"

8.5 Mastering Overview

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 America Corporation. However, it is possible that a robot might lose its mastering data and need to be remastered.

The robot axes are controlled by a closed loop servo system. The controller outputs a command signal to drive each motor. A feedback device mounted on the motor, called a serial pulse coder, sends a signal back to the controller. During robot operation, the controller constantly analyzes the feedback signal, and modifies the command signal to maintain proper location and velocity of the end effector at all times.

For the robot to move accurately to recorded positions, the controller has to "know" the position of each axis. It does this by comparing the serial pulse coder reading during operation with a reading taken at a known mechanical reference point on the robot. Mastering records the serial pulse coder reading at a known mechanical reference point. This mastering data is stored with other user data in the controller and maintained by a backup battery when the controller is shut down.

When the controller is shut down under normal conditions, each serial pulse coder's present reading is maintained in the pulse coder by backup batteries on the robot. (These batteries might be located in the controller on P-series robots.) When the controller is turned on, the controller requests the stored reading from the serial pulse coder. When the controller receives the pulse coder reading, the servo system can operate normally. This process is called calibration. Calibration occurs automatically each time the controller is turned on.

8.5.1 Verify Mastering

Prior to creating programs and recording points it is necessary to ensure the robot is properly mastered. If the robot master is off prior to recording positions, when the mastering values are corrected all of the program points will be off by the change in the mastering values.

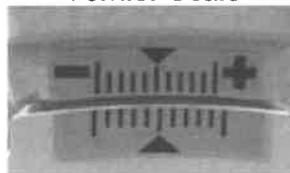
Procedure 8-5 Verifying the Robot is Mastered

- 1 Utilizing the position screen joint jog each axis to zero degrees
- 2 Check the alignment marks for each axis to ensure they are aligned.
 - a The robot will be equipped with one of the styles of alignment marks shown below.

Old Style Scribe Mark



Vernier Scale



New Style Scribe Mark



8: Initial Setup

- 3 If the marks are aligned at zero degrees the robot master is verified.
- 4 If the marks do not align at zero degrees, mastering must be performed.

8.6 Chapter Review

1. **What must be done after setting Software Axis Limits?**
 - A Cycle Power
 - B Nothing
 - C Jog to each extreme of the limit to finalize
 - D Software Axis Limits are not supported by FANUC
2. **How do you recover from violating an axis limit?**
 - A Cycle power.
 - B Press Reset.
 - C Jog the opposite direction.
 - D Change the limit to allow for motion.
3. **Robot payload information is comprised of _____, COG, and Inertia.**
 - A Weight
 - B Height
 - C Span
 - D Mass
4. **If a robot loses its master, it will run just fine.**
 - A True
 - B False

8: Initial Setup

Lab 6 Axis Limits

Student Name: _____

Assignment: The student will:

- Set Axis Limits
- Verify Limit Settings

Condition: Given a FANUC Robot and a controller

Step: 1 Display current axis limits and record the information below. (See Section 8.3, pg. 92)

Axis	Lower	Upper
1	_____	_____
2	_____	_____
3	_____	_____
4	_____	_____
5	_____	_____
6	_____	_____

2 Change the axis limits.



Never increase the range of motion from the default values, damage to the robot or injury to personnel may occur.

3 Verify the limits have taken affect by jogging into the limit. What message was triggered when you hit the limit?

4 Recover from the limit.

5 Reset the axis limits to original values.

Completed:

Instructor: _____

Lab 7

Verify Mastering

Student Name: _____

Assignment: The student will:

- Jog the robot to verify correct mastering
-

Condition: Given a FANUC Robot and a controller

- Step:**
- 1 Jog the robot to align the Vernier Scales / Scribe Marks. (See Section 8.5.1 Verify Mastering, pg. 97)
 - 2 Verify the robot master is correct.

Completed:

Instructor: _____

Lab 8 Set Payload Data

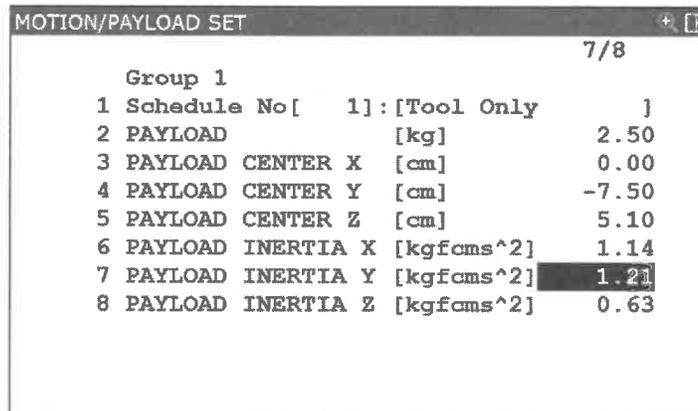
Student Name: _____

Assignment: The student will:

- Create a Payload Schedule
- Set a Payload Schedule

Condition: Given a FANUC Robot and a controller

- Step: 1 Directly enter Payload Data for Schedule [1]. (See Section 8.4.1 Payload Setup, pg. 94)

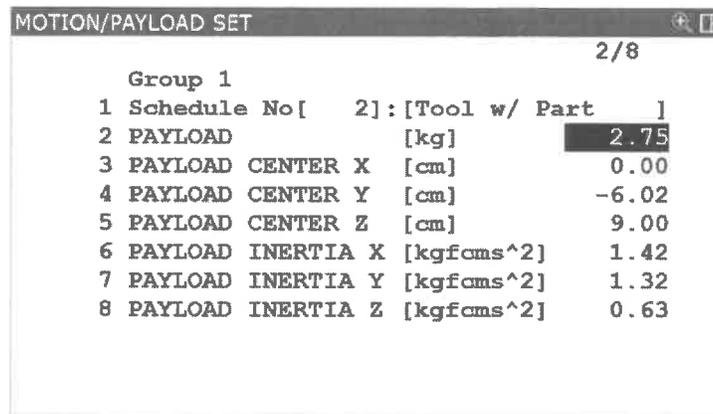


MOTION/PAYLOAD SET 7/8

Group 1

1	Schedule No[1]:[Tool Only]	
2	PAYLOAD [kg]	2.50
3	PAYLOAD CENTER X [cm]	0.00
4	PAYLOAD CENTER Y [cm]	-7.50
5	PAYLOAD CENTER Z [cm]	5.10
6	PAYLOAD INERTIA X [kgfcms^2]	1.14
7	PAYLOAD INERTIA Y [kgfcms^2]	1.21
8	PAYLOAD INERTIA Z [kgfcms^2]	0.63

- 2 Directly Enter Payload Data for Schedule [2].



MOTION/PAYLOAD SET 2/8

Group 1

1	Schedule No[2]:[Tool w/ Part]	
2	PAYLOAD [kg]	2.75
3	PAYLOAD CENTER X [cm]	0.00
4	PAYLOAD CENTER Y [cm]	-6.02
5	PAYLOAD CENTER Z [cm]	9.00
6	PAYLOAD INERTIA X [kgfcms^2]	1.42
7	PAYLOAD INERTIA Y [kgfcms^2]	1.32
8	PAYLOAD INERTIA Z [kgfcms^2]	0.63

- 3 Using F5, SETIND set Schedule [1] as the active payload schedule.

Completed:

Instructor: _____

9 ALARMS

9.1 Learning Objectives

- Diagnose Faults and Messages
- Recover from common errors

9.2 Fault and Message Diagnosis

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 over travel has occurred.

Most errors can be cleared by correcting the problem and pressing the Reset Key on the teach pendant or SOP. However, blindly pressing reset to clear an alarm may not actually solve the issue. It is best to gather an understanding of what the alarm means and recommended steps to correct the issue.

If troubleshooting it is usually best to view the Active alarms in the alarm log or history. This can aid in determining if there are any other active alarms as well and can aid in sorting out what is the actual problem.

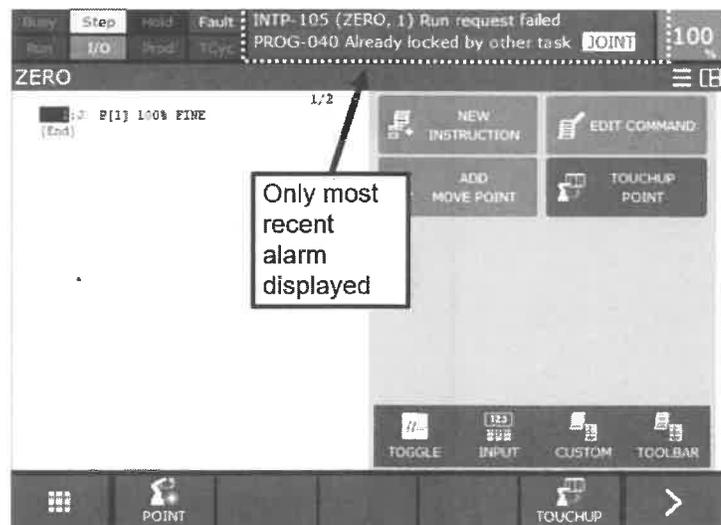
Another useful tool to determine the cause and remedies of alarms is the diagnostic tool. Diagnosing an alarm of message gives the user the possible cause and possible remedies to fix the issue. This can be useful for an operator to determine if they need to notify maintenance to correct the issue.

9.2.1 Displaying the Alarm Log

The alarm log is an invaluable tool for use in understanding active alarms. The teach pendant only displays the most recent alarm along the title bar of the pendant, if there are multiple alarms though they will not be displayed.

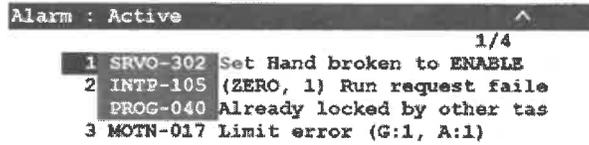
The alarm screen can be forced to display all of the active alarms whenever an alarm is triggered, however for troubleshooting purposes this may not be desirable.

A better alternative is to manually display the Alarm screen as needed.



9: Alarms

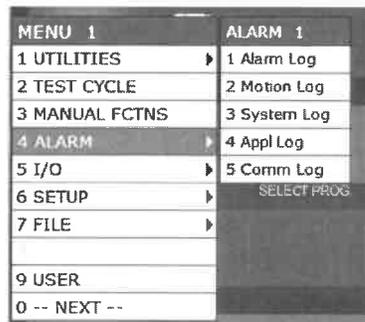
A benefit of displaying the Alarm screen is to determine if there are multiple alarms active at one time. As seen on the previous page only one alarm is displayed along the top title bar of the iPendant, however in the image below we can see there are a number of alarms active.



Determining if there are any other alarms can aid in deciding which alarms needs to be corrected first. To identify how to fix the issues these alarms can be easily diagnosed.

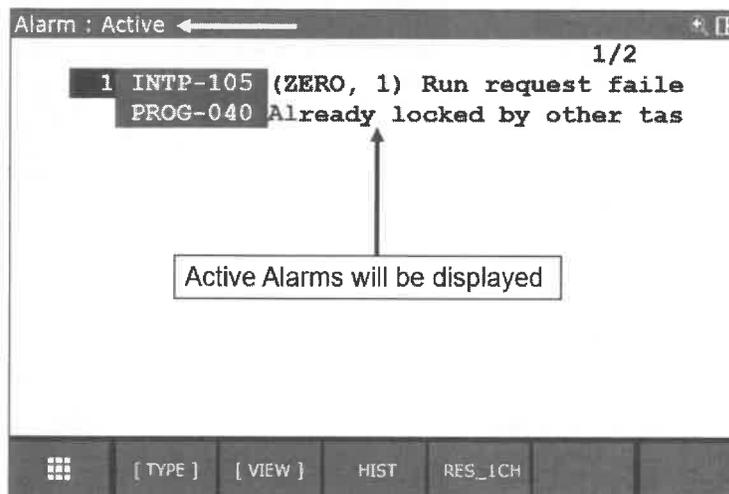
Procedure 9-1 Displaying the Alarm Screen

- 1 Press Menu



- 2 Cursor down to Alarms and press Enter or Press the number 4 on the keypad.

- 3 You will see a screen similar to below.

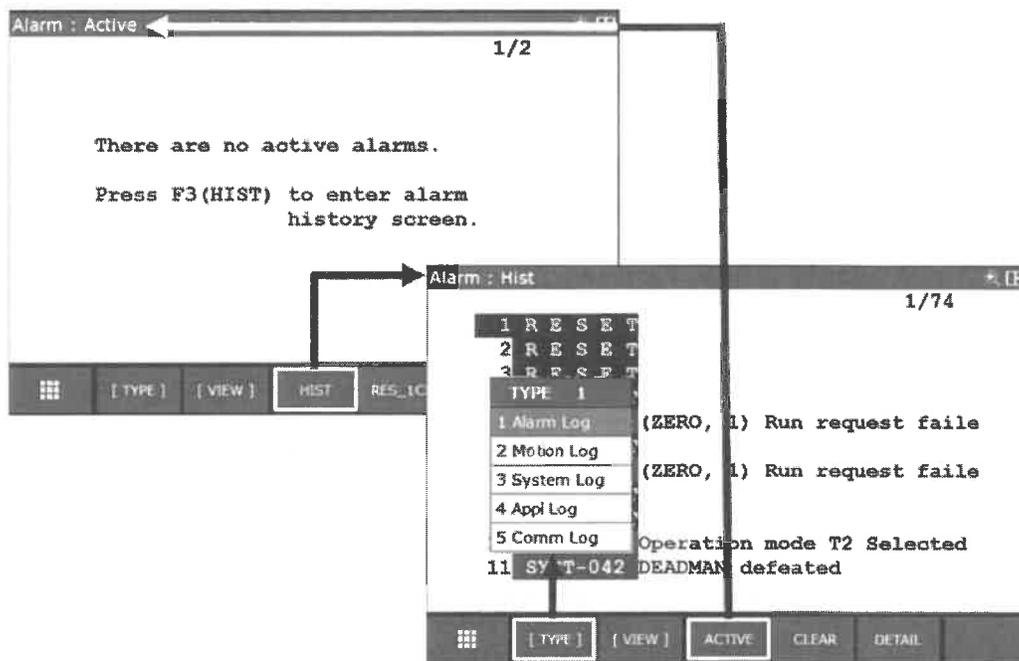


9.2.2 Displaying the Alarm History

From the Alarm screen you can view current alarms and also gain access to the alarm history. The alarm history stores the 100 most recent alarms. These can be used to determine what has been occurring on the robot system. The history of errors and messages can also be sorted based on alarm type and show the most recent 50 alarms of each type.

To navigate the history and sort alarms refer to the below:

Procedure 9-2 Access Alarm History:	Procedure 9-3 Sort by alarm type:	Procedure 9-4 Return to Active Alarms:
<ol style="list-style-type: none"> 1 Navigate to the Alarm Screen. 2 Press F[3] HISTORY 	<ol style="list-style-type: none"> 1 Navigate to the Alarm Screen. 2 Press F[1] [TYPE]. 3 Choose specific type from the list. 	<ol style="list-style-type: none"> 1 From a History Screen. 2 Press F3 Active.



9.2.3 Diagnosing a Fault

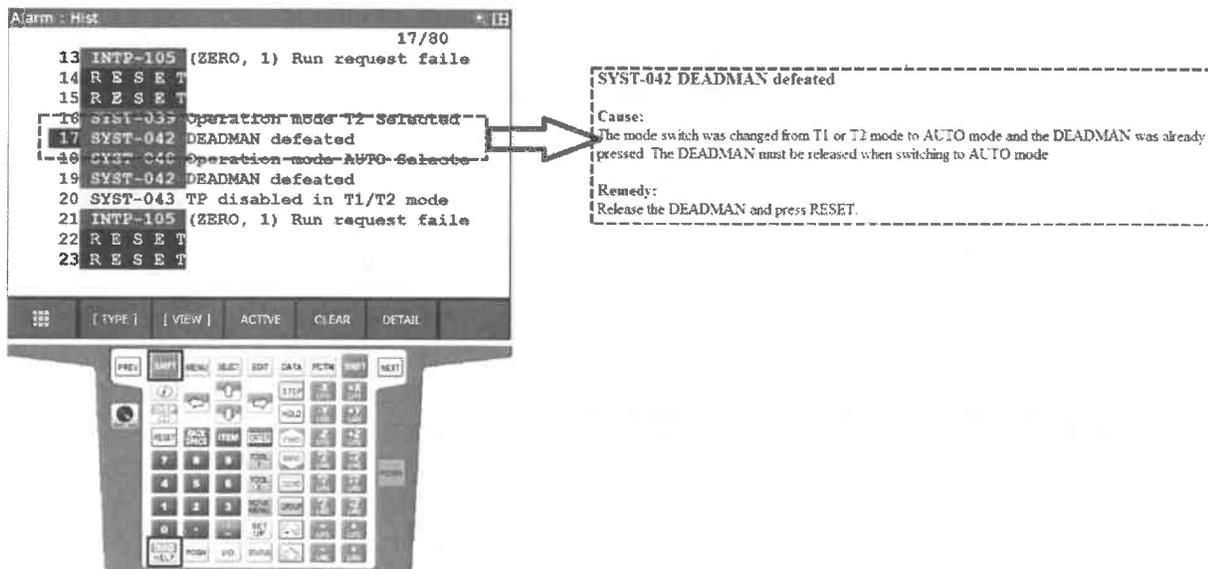
A fault can be diagnosed from multiple locations. If there is only one fault any time you press Shift and Diagnose it will diagnose the current fault. A better approach is to navigate to the Alarm and view the faults there. This will allow the operator to ensure they are aware of all faults and have the ability to diagnose a specific fault.

9: Alarms

From the alarm screen a fault or message can be diagnosed from the active screen as well as any history screens.

Procedure 9-5 Diagnosing a Fault or Message

- 1 Navigate to the appropriate screen, active alarms or history if necessary.
- 2 Place the cursor next to the target fault or message.
- 3 Press Shift and DIAG together



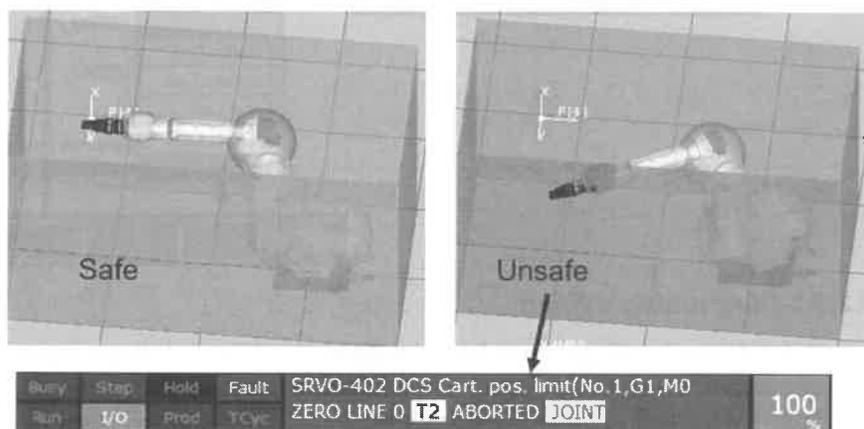
- 4 To leave the diagnosis screen press PREV.

9.3 Dual Check Safety (DCS) Option

Dual Check Safety (DCS) Position/Speed Check features check the speed and position data of motors with two independent CPUs in the robot controller. These functions can detect position and speed errors immediately and shut down the motor power.

Procedure 9-6 Recover from a DCS Position violation:

- 1 Press Shift and Reset
- 2 Jog away from the limit



9.4 Singularity

Singularity occurs in an X-Y-Z/Cartesian coordinate only. For most six-axis industrial robots, wrist singularity occurs when the axes of Joints 4 and 6 are aligned. When axis 5 (J5) approaches 0 degrees, the robot encounters an infinite combination of axis 4 and axis 6 positions to achieve a location.

Said differently, singularity occurs when an infinite number of joint angles generate the same set of Cartesian data (J4, J5 and J6 on a 6 axis robot). It is important to keep in mind singularity is not a unique FANUC issue, it is a six axis robot issue.

 **NOTE:** Joint jogging is not effected by Singularity errors.



Procedure 9-7 Recovering from Singularity

- 1 Press the COORD key to select Joint
- 2 Move Joint 5 "+" or "-" 10 degrees
- 3 Return to prior Cartesian Coordinate motion or continue to jog the robot.

9.5 Chain Failure Detection Error

The robot controller may be equipped with a dual chain Control Reliable safety circuit. The circuit is designed such that a single component failure within the system does not prevent the stopping action from taking place but will prevent successive system cycle or movement until that failure has been corrected.

A Single Chain Failure Detection fault will be set if one safety chain is in an Emergency Stop condition and the other is not in an Emergency Stop condition.

If this occurs the teach pendant will display one of the following faults:

- SRVO-230 Chain 1 (+24v abnormal)
- SRVO-231 Chain 2 (0v abnormal)

WARNING

If you reset the chain failure fault without fixing the cause of it, the same alarm will occur, but the robot can move until the alarm occurs again. Be sure to fix the cause of the chain failure before you continue. Otherwise, you could injure personnel or damage equipment.

Procedure 9-8 Resetting a Chain Failure Alarm

- 1 Correct the cause of the alarm.
- 2 Create and then release a chain failure that is different from the original chain failure. An

9: Alarms

example of this would be to push the EMERGENCY STOP button on the teach pendant and then press RESET.

- 3 Press MENU.
- 4 Select ALARMS.
- 5 Press F4 RES_1CH



- 6 Verify YES by pressing F4.

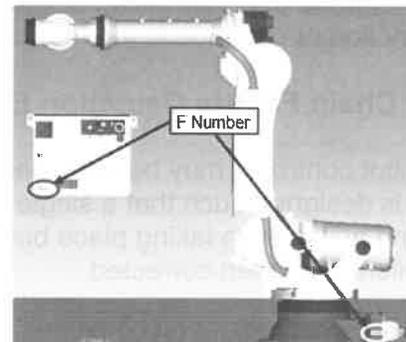


- 7 Press RESET to clear the fault from the Active Alarm screen.

9.6 Version Identification

9.6.1 Robot F-Number

FANUC America Corporation provides an F number which is a unique number for the robot's identification. The F number provides information such as controller type, robot model, installed options and sale information. The number is located at the base of the robot and on the controller cabinet.



9.6.2 Version Identification Status

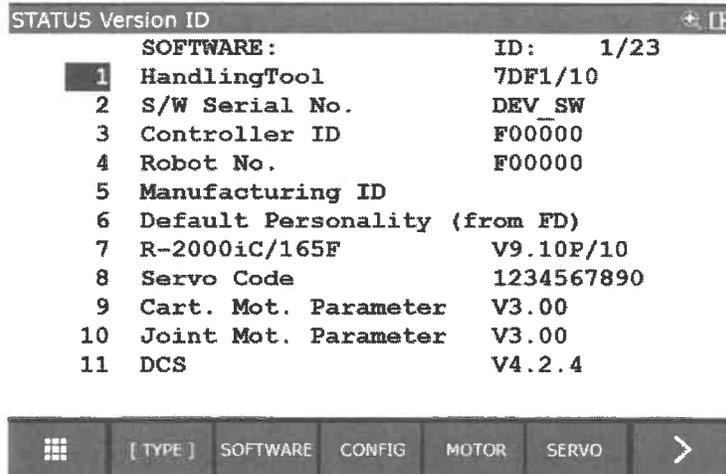
The STATUS Version ID screen displays information specific to your controller. Use this information when you call the FANUC America Corporation Hotline if a problem occurs with your controller. You cannot change the information displayed on this screen.

Procedure 9-9 Displaying the Version ID Information

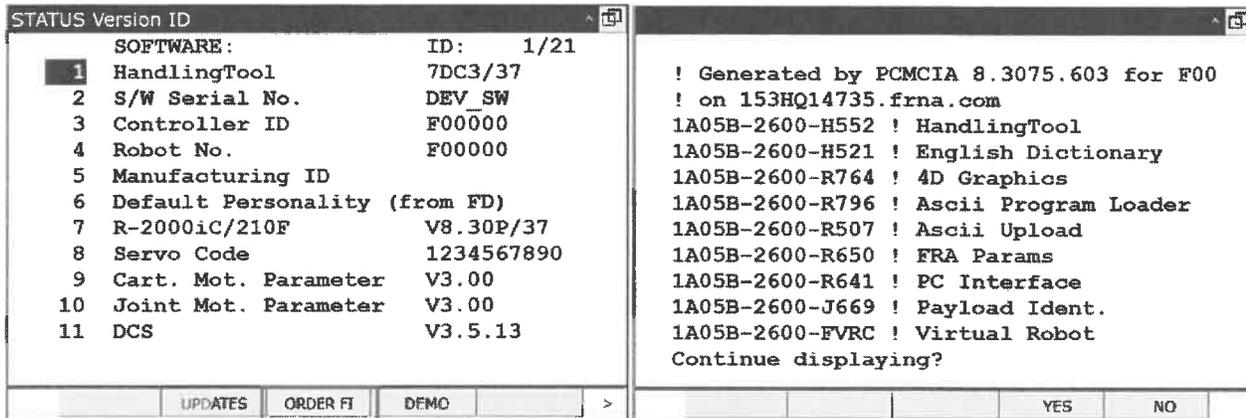
- 1 Press Menu, -0- Next, and cursor down to Status.
- 2 On the Status fly out menu cursor down to Version ID and press Enter.

MENU 2	STATUS 1	US 2
1 SELECT	1 Axis	
2 EDIT	2 Version ID	Program
3 DATA	3 Stop Signal	Pointer
4 STATUS	4 Exec-hist	Notifications
5 4D GRAPHICS	5 Memory	
6 SYSTEM	6 Robot Condition	
7 USER2	7 Prg Timer	
8 BROWSER	8 Sys Timer	
	9 Condition	
0 -- NEXT --	0 -- NEXT --	0 -- NEXT --

3 You should see a screen similar to below.



4 Pressing NEXT and then F3 ORDER FI (Order Fill) will display specific software features and options that are loaded on the controller. The screen is similar to below.



9.7 Chapter Review

1. When a DCS error occurs, what two keys do you press in order to clear the fault?

_____ and _____

2. Robot F Number and Version ID information can helpful for what reasons?

3. Singularity error can be corrected by moving? (Circle the correct answer)

- A Joint 2 (+/-) 10 degrees
- B Joint 3 (+/-) 15 degrees
- C Joint 5 (+/-) 10 degrees
- D Joint 6 (+/-) 15 degrees

Lab 9
Diagnosing a Fault

Student Name: _____

- Assignment: The student will:
- Navigate to the Alarm Screen
 - Diagnose a Fault

Condition: A FANUC robot and controller.

- Step: 1 Navigate to the Alarm Screen. (See Section 9.2.1 Displaying the Alarm Log, pg. 103)
- 2 Display the alarm history. (See Section 9.2.2 Displaying the Alarm History, pg. 105)
- 3 Diagnose any fault listed on the Alarm Screen. (See Section 9.2.3 Diagnosing a Fault, pg. 105)
- a In the space provided write down the cause and remedy of this alarm

Cause:

Remedy:

Completed:

Instructor: _____

Lab 10

Viewing the Version Identification Information

Student Name: _____

Assignment: The student will:

- Determine the Version ID Information.
-

Condition: Given a FANUC Robot and a controller

- Step:**
- 1 Display the Version ID information. **(See Section 9.6.2 Version Identification Status, pg. 108)**
 - 2 What is the Software Edition Number? _____.
 - 3 What is the S/W Serial Number? _____
 - 4 What is the controller F Number? _____
 - 5 What is the Robot Model? _____

Completed:

Instructor: _____

10 FRAMES

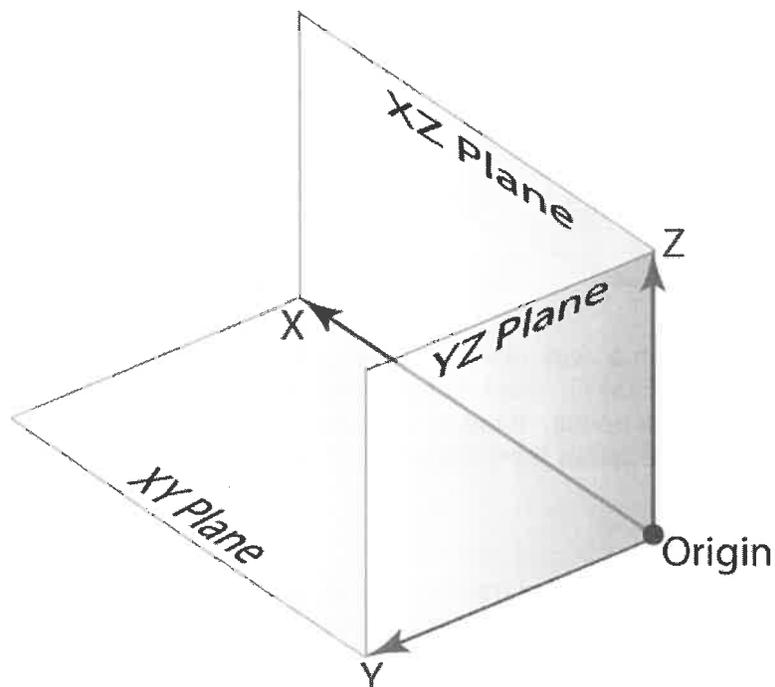
10.1 Learning Objectives

- Understand a 3D Cartesian Coordinate System
- Learn the different Frame Types
- Create Tool Frames
- Create User Frames
- Create Jog Frames
- Save Frame Data

10.2 Frames Overview

As discussed in Chapter 6, the FANUC Application Software utilizes X, Y, Z, W, P, and R to move throughout the cell and record positional data, this setup is known as a 3 Dimensional Cartesian coordinate system.

A 3D Cartesian coordinate system is a set of three planes at right angles to each other. The point at which all three planes intersect is the origin of the system. FANUC typically refers to such a system as a Frame. In the robot system, the intersecting edges of the planes are the X, Y, and Z axes of the frame.



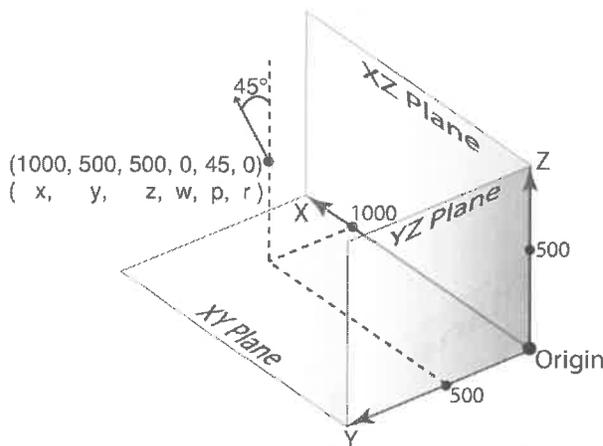
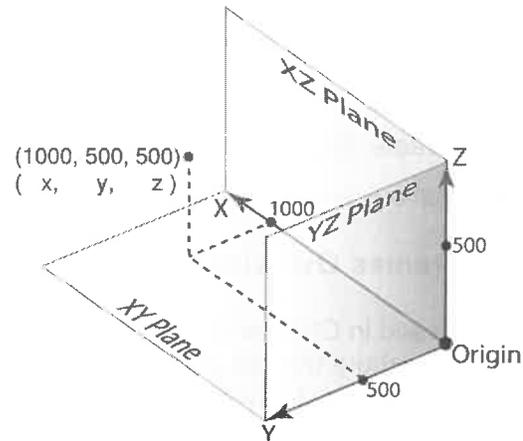
10: Frames

10.2.1 How are Frames Used

Frames are used to describe the **location** and **orientation** of a position in three-dimensional space. 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 as three dimensions, which, in the robot, are measured in millimeters from the origin in the X, Y, and Z directions.

As shown in the image to the right, 1000,500,500 means the position is 1000mm in the X direction, 500mm in the Y direction, and 500mm in the Z direction from the origin.



The **orientation** of a robot position is measured in degrees of rotation about the X, Y, and Z axes.

As shown in the image to the left, 0,45,0 means that the position is rotated +45 degrees about the Y axis and is not rotated about the X or Z axes.

One of the benefits to defining frames is to establish a reference point on a work piece such as a fixture or pallet and also a reference point on the End of Arm Tooling (EOAT). Once established, the positional data mentioned above will be relative to these reference points. If the work piece or tooling moves, the program can be edited by adjusting the frame rather than adjusting all positional data in each program.

10.3 Frame Types

The robot utilizes five different kinds of frames. Each different kind of frame makes certain tasks; jogging, offsetting, programming, etc. easier to perform. The kinds of frames are:

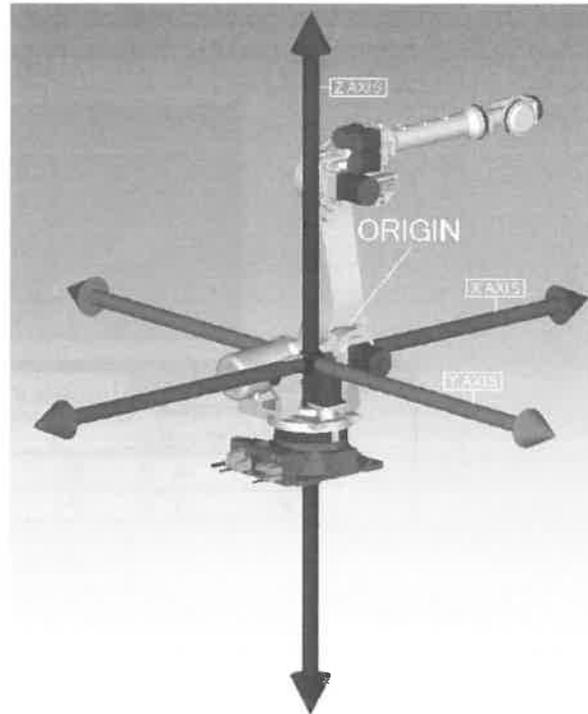
- World Frame
- Tool Frame
- User Frame
- Jog Frame
- Cell Frame

NOTE: Cell Frame is used in the 3D graphics and not covered further in this manual.

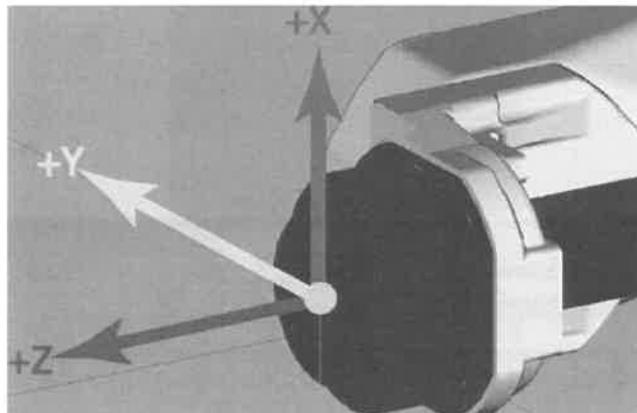
10.4 World Frame

WORLD is the default frame used to describe the location and orientation of the Tool Center Point (TCP). The default location of the TCP is the center of the faceplate. The TCP location can be changed once tooling has been added to the robot. When recording positional data for a program the robot measures from the WORLD origin to the tooling and is represented in X, Y, Z, W, P, and R coordinate values.

- The WORLD Frame is the default frame of the robot. It cannot be changed by the user.
- It is the basis for all taught positions and all taught frames within the robot.
- The origin of the world frame is located on the centerline of the J1-axis at the height of the centerline of the J2-axis.
- The location of this origin never changes.
- The orientation of the WORLD frame can change from improper mastering.
- When jogging in WORLD, the servo motors will maintain a linear (straight) movement with respect to the tool.

**10.5 Tool Frames**

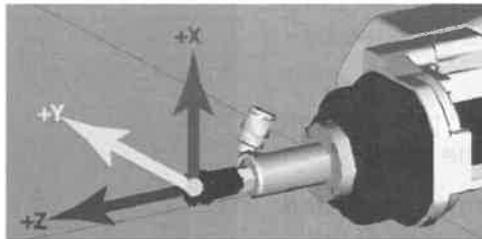
Tool frame is used to describe the orientation and location of the physical tool on the robot. By default, the tool frame has its origin at the robot faceplate.



10: Frames

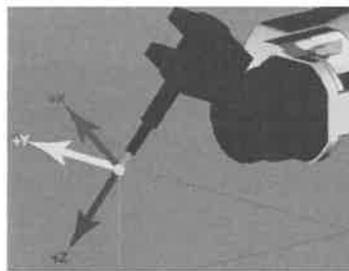
When you set up a Tool Frame (commonly referred to as the Tool Center Point (TCP)), you move the origin of this frame from the robot faceplate to the point on the applicator, gun, torch, or tool at which the painting, welding, sealing, handling, or other application work is to be done.

Once the Tool Frame has been defined the positional data in your programs will now be relative to the distance from the World origin to the defined center point of the tool. This will make your programming data more accurate to where the work is actually being performed. Also, when conducting Cartesian Interpolated motion, the path will be based on the TCP and move the defined TCP in a linear or circular manner depending on program commands.



SETUP Frames		Location Values		int 2/4	
Tool Frame					
Frame Number:	3				
X:	0.0	Y:	0.0	Z:	75.0
W:	0.0	P:	0.0	R:	0.0
Orientation Values					

The Tool Frame data is the physical distance that the center point of the frame is from the faceplate of the robot relative to X, Y, and Z. The W, P, and R values of the tool frame identify the orientation of the frame. From the image above we can see the Tool Frame maintained the default orientation which is Z+ straight out from the faceplate. As seen below, if we were to add a positive or negative value in the W, P, and/or R fields of the frame the orientation would have rotated about the specified plane(s).



SETUP Frames		Location Values		int 2/4	
Tool Frame					
Frame Number:	3				
X:	0.0	Y:	0.0	Z:	75.0
W:	0.0	P:	-45.0	R:	0.0
Orientation Values					

When working with Tool Frames keep in mind:

- When no data is defined the Tool Frame location is the center of the faceplate.
- The Tool Frame data is a physical measurement back to the center of the faceplate.
- Once defined the Tool Frame is commonly referred to as a Tool Center point (TCP).
- Robots with multiple tools can have multiple TCP's, default of ten schedules.
 - R30iB Controllers can have up to 29 Tool Frames
 - R30iB Plus Controllers can have up to 253 Tool Frames
- When selecting Tool as the jog method the active TCP becomes the focal point while jogging.

10.6 Creating a Tool Frame

There are six standard ways to define a Tool Frame:

- Two Point +Z This function can set TCP for robots which cannot tilt their tools to the XY plane in the world frame (i.e. 4 axes robots).
- Three Point Defines the location of the tool frame when the values cannot be measured and directly entered. **See 10.6.1 Three Point Method.**
- Four Point Defines the precise location of the tool frame by utilizing the feedback provided by this method on how well the approach positions are taught. **See 10.6.2 Four Point Method.**
- Six Point (XZ) Define the location and orientation of the tool frame using the X and Z planes when the values cannot be measured and directly entered. **See 10.6.3 Six Point (XY / XZ) Method.**
- Six Point (XY) Defines the location and orientation of the tool frame using the X and Y planes when the values cannot be measured and directly entered. **See 10.6.3 Six Point (XY / XZ) Method.**
- Direct Entry The direct entry method provides for direct recording and numerical entry of the frame position. **See 10.6.4 Direct Entry Method.**

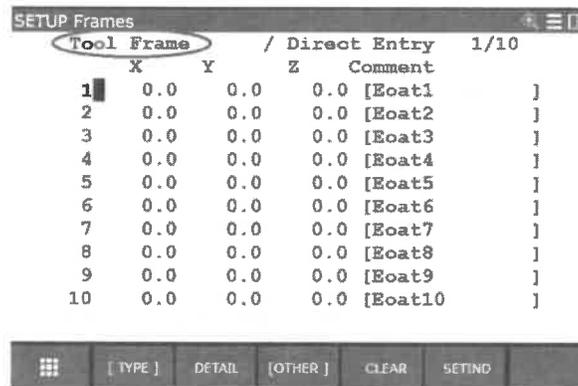
 **NOTE:** The two point +Z is not covered in this manual. Refer to the specific application software manual for further details if necessary.

10.6.1 Three Point Method

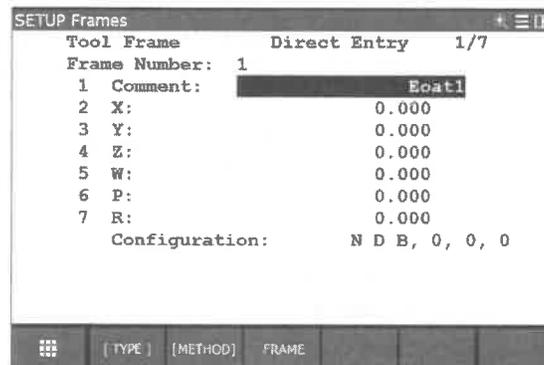
Use the three point method to define the location of the tool frame when the values cannot be measured and directly entered. The three approach points must be taught with the tool touching a common point **from three different approach** directions.

Procedure 10-1 Defining a Tool Frame using the Three Point Method

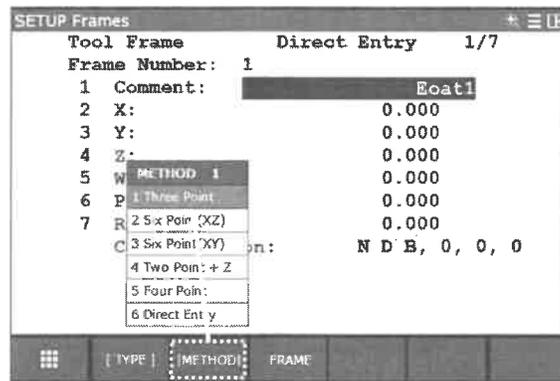
- 1 Press Menu
 - a Cursor down to SETUP
 - b Press the right arrow and cursor down to Frames
 - c Press Enter
 - d You should see a screen similar to below.



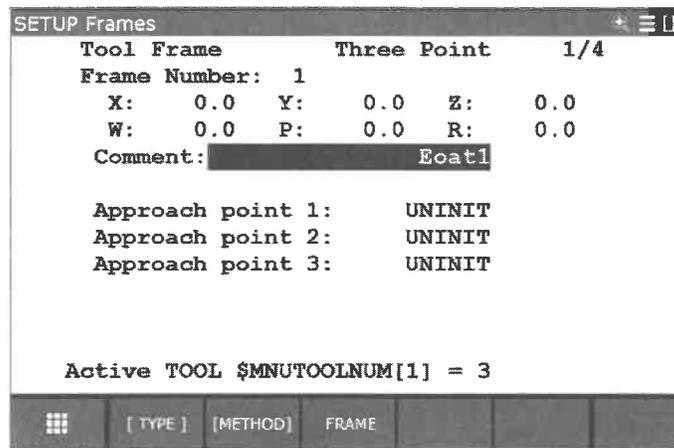
- 2 Look in the upper left hand corner, if something other than Tool Frame is displayed, press F3 [OTHER] and select Tool Frame. If OTHER is not displayed press Prev.
- 3 Place your cursor next to The Tool Frame to define and press Enter or F2 DETAIL.



- 4 Press F2 [METHOD] highlight Three Point and press Enter.

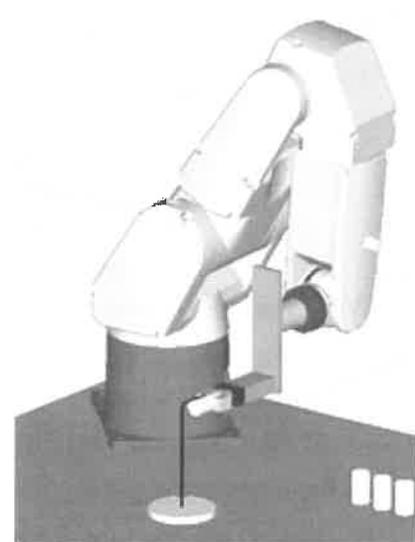
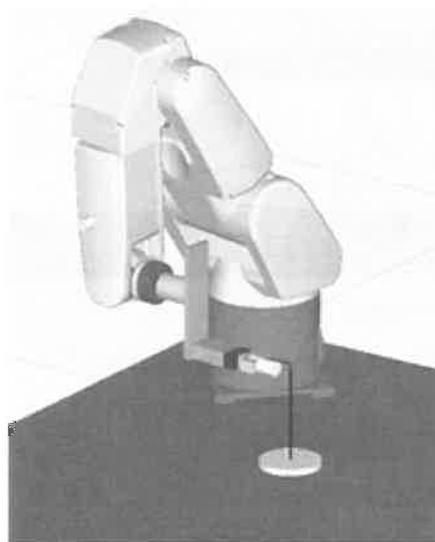
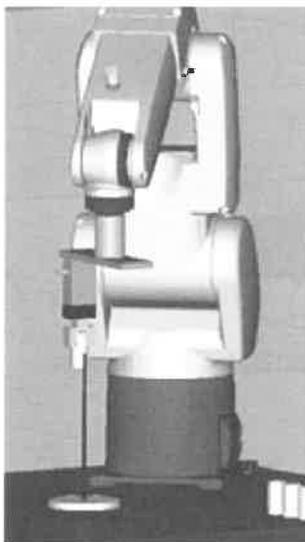


5 You should see a screen similar to below.



6 Add a comment to associate the Tool Frame with a specific tool.

7 Record Approach Points, the objective is to record the approach points from three different angles. The figures below show examples of how the approach points may be recorded



- 8 Cursor to the line labeled Approach point 1.
 - a Jog the robot so that the tool tip touches a fixed reference point
 - b With the robot properly positioned hold down Shift and F5 RECORD to record the position.
- 9 Cursor to the line labeled Approach point 2.
 - a Jog the robot so that the tool tip touches the fixed reference point but this time approach the point from a different angle.

NOTE: Large rotations, near 90° but no more than 360°, give the best results. However, smaller rotations can be used if motion is restricted by cabling or other attachments.

- b With the robot properly repositioned hold down the SHIFT key then press F5 RECORD.

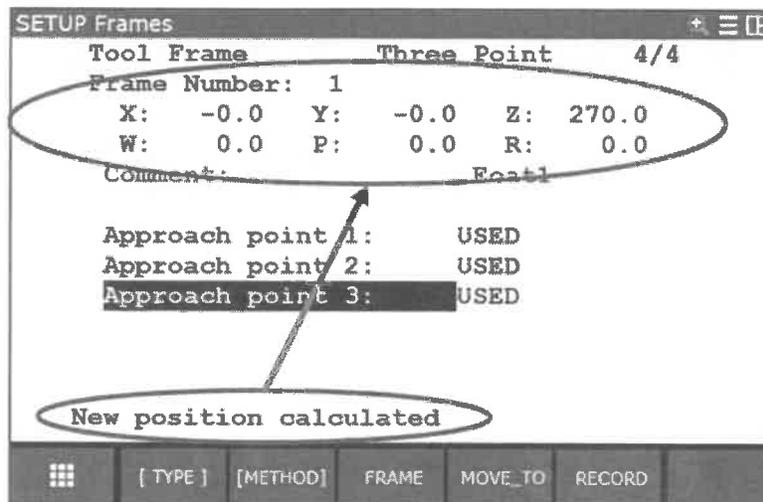
- 10 Cursor to the line labeled Approach point 3.

- a Jog the robot so that the tool tip touches the fixed reference point but this time approach the point from a different angle.

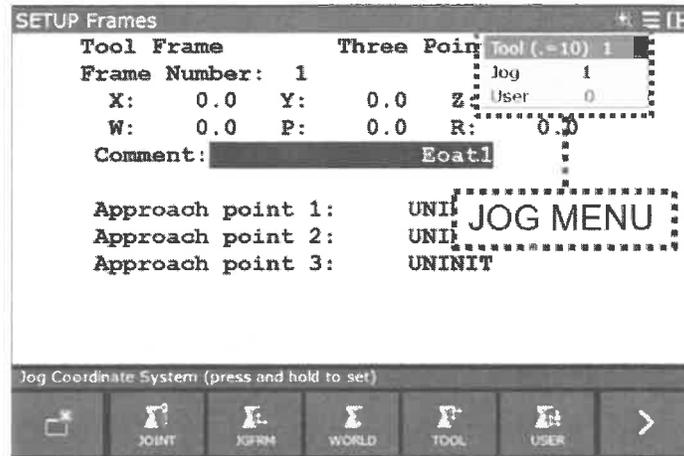
NOTE: Large rotations, near 90° but no more than 360°, give the best results. However, smaller rotations can be used if motion is restricted by cabling or other attachments.

- b With the robot properly repositioned hold down the SHIFT key then press F5 RECORD.

- 11 After recording the final position the Frame data will be calculated. You should see a screen similar to below.



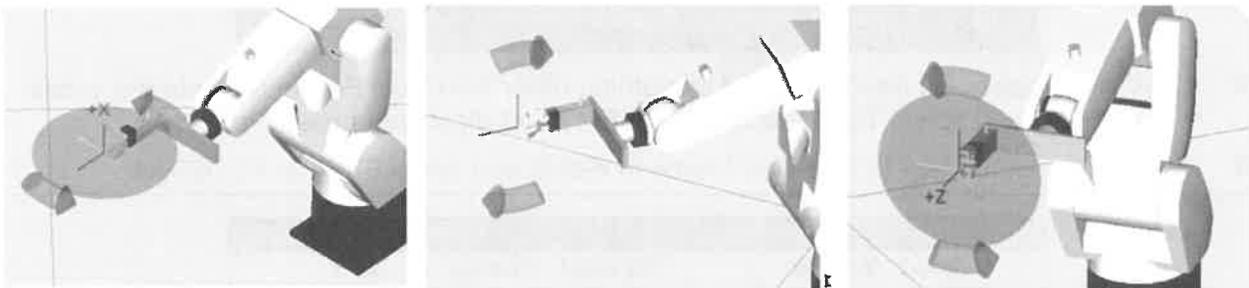
- 12 With the new Frame calculated proof the values.
 - a Measure back to the face plate from the newly defined TCP and ensure the numbers make sense.
- 13 Test the newly defined Frame by jogging as well.
 - a Hold Shift and press COORD.
 - b On the Jog Menu enter the value of the Frame you just created on the Tool line.



c Ensure Tool is the selected Jog Method, press COORD until it is displayed.



d Using the rotation keys, jog the robot about the X, Y, Z coordinate axes. If your frame is taught correctly the rotational movements of the robot with pivot around the point in space that you defined for your Tool Frame and not drift away from it.



10.6.2 Four Point Method

Use the four point method to define the precise location of the tool frame by utilizing the feedback provided by this method on how well the approach positions are taught. This method requires four approach points to be taught with the tool touching a common point from four different approach positions.

This method indicates the approach position with maximum TCP error so that user can correct that position and re-compute the tool frame. This will reduce the effect of user's teaching errors incorporated in the computation of the tool frame. Adjust and re-teach the approach positions that are slightly off iteratively until the tool frame computed is of desired accuracy on the basis of the TCP error feedback.

Procedure 10-2 Defining a Tool Frame using the Four Point Method

- 1 Press Menu
 - a Cursor down to SETUP
 - b Press the right arrow and cursor down to Frames
 - c Press Enter
 - d You should see a screen similar to below.

SETUP Frames					
Tool Frame	/ Direct Entry			1/10	
	X	Y	Z	Comment	
1	0.0	0.0	0.0	[Eoat1]
2	0.0	0.0	0.0	[Eoat2]
3	0.0	0.0	0.0	[Eoat3]
4	0.0	0.0	0.0	[Eoat4]
5	0.0	0.0	0.0	[Eoat5]
6	0.0	0.0	0.0	[Eoat6]
7	0.0	0.0	0.0	[Eoat7]
8	0.0	0.0	0.0	[Eoat8]
9	0.0	0.0	0.0	[Eoat9]
10	0.0	0.0	0.0	[Eoat10]

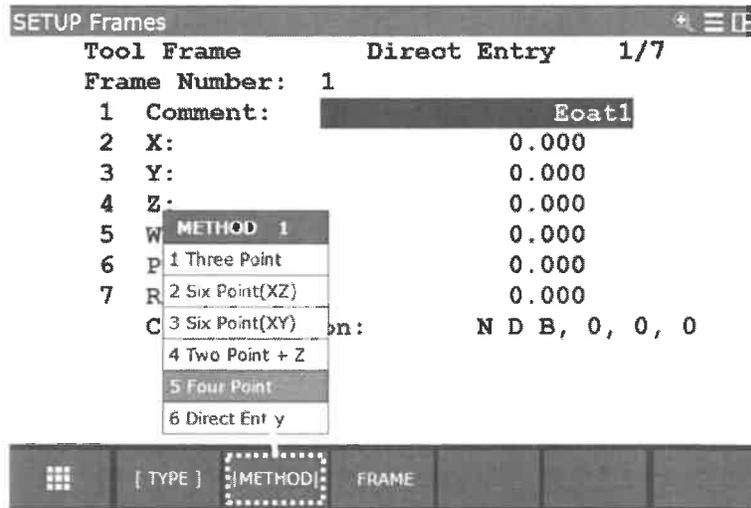
[TYPE] [OTHER] CLEAR SETIND

- 2 Look in the upper left hand corner, if something other than Tool Frame is displayed, press F3 [OTHER] and select Tool Frame. If OTHER is not displayed press Prev.
- 3 Place your cursor next to The Tool Frame to define and press Enter or F2 DETAIL.

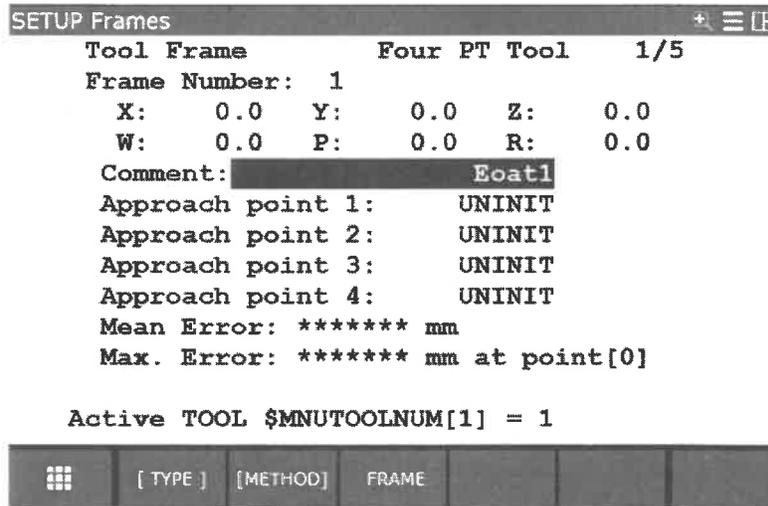
SETUP Frames			
Tool Frame	Direct Entry		1/7
Frame Number:	1		
1 Comment:	Eoat1		
2 X:	0.000		
3 Y:	0.000		
4 Z:	0.000		
5 W:	0.000		
6 P:	0.000		
7 R:	0.000		
Configuration:	N D B, 0, 0, 0		

[TYPE] [METHOD] FRAME

- 4 Press F2 [METHOD], highlight Four Point and press Enter.

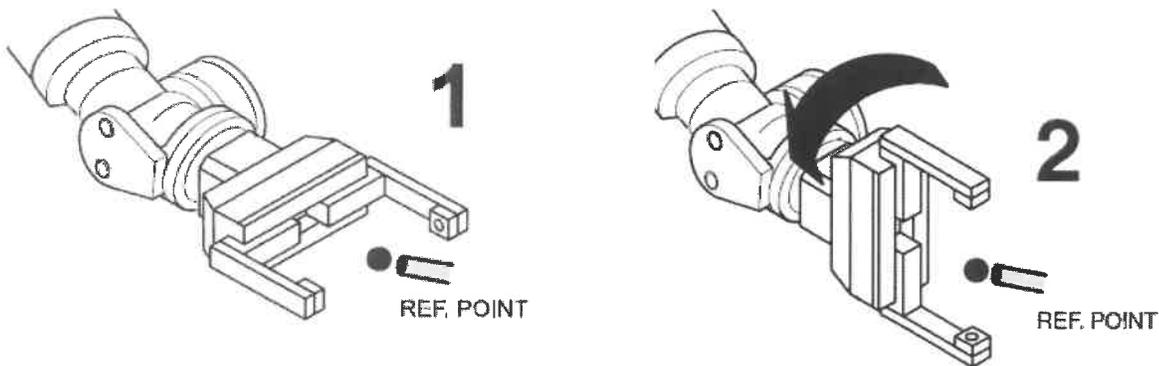


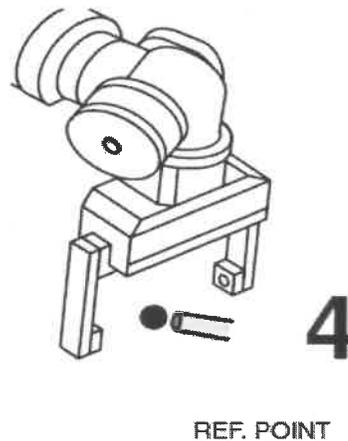
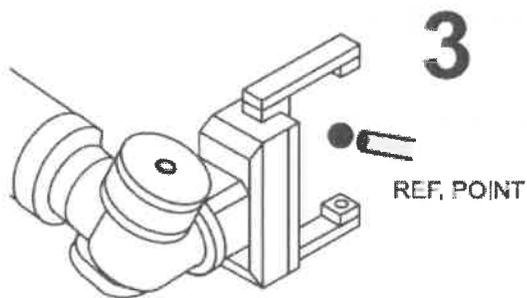
5 You should see a screen similar to below.



6 Add a comment to associate the Tool Frame with a specific tool.

7 Record Approach Points, the objective is to record the approach points from four different angles. The figures below show examples of how the approach points may be recorded





- 8 Cursor to the line labeled Approach point 1.
 - a Jog the robot so that the tool tip touches a fixed reference point
 - b With the robot properly positioned hold down Shift and F5 RECORD to record the position.
- 9 Cursor to the line labeled Approach point 2.
 - a Jog the robot so that the tool tip touches the fixed reference point but this time approach the point from a different angle.

NOTE: Large rotations, near 90° but no more than 360°, give the best results. However, smaller rotations can be used if motion is restricted by cabling or other attachments.

- b With the robot properly repositioned hold down the SHIFT key then press F5 RECORD.

- 10 Cursor to the line labeled Approach point 3.
 - a Jog the robot so that the tool tip touches the fixed reference point but this time approach the point from a different angle.

NOTE: Large rotations, near 90° but no more than 360°, give the best results. However, smaller rotations can be used if motion is restricted by cabling or other attachments.

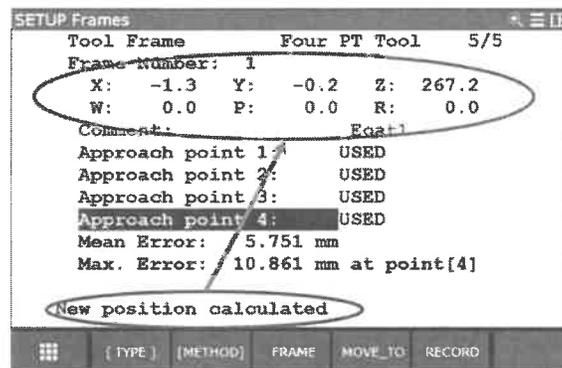
- b With the robot properly repositioned hold down the SHIFT key then press F5 RECORD.

- 11 Cursor to the line labeled Approach point 4.
 - a Jog the robot so that the tool tip touches the fixed reference point but this time approach the point from a different angle.

NOTE: Large rotations, near 90° but no more than 360°, give the best results. However, smaller rotations can be used if motion is restricted by cabling or other attachments.

- b With the robot properly repositioned hold down the SHIFT key then press F5 RECORD.

- 12 After recording the final position the Frame data will be calculated. You should see a screen similar to the following page.



- 13 You will see the message, “New position calculated” and the tool frame computed will be displayed along with the mean and max errors of all four approach points.
- 14 If the mean and max TCP errors are within desired limits, the new tool frame is ready to use.

CAUTION

When the mean error is very small, it is possible that three of the four recorded positions are inaccurate (relatively speaking) resulting in the correct position having the largest deviation from the mean. Using a precise tool and a reference point for teaching should overcome this.

- 15 If the user wants better accuracy, correct the approach position corresponding to max error and re-compute the tool frame.
 - a Move the cursor to the Approach point indicated as having the max error.
 - b Press and hold the SHIFT key and press F4, MOVE_TO.
 - c Adjust the position to correct any positional errors.
 - d Press and hold the SHIFT key and press F5, RECORD.
 - e Tool frame is computed again.
 - f Repeat these steps if more accuracy is needed.

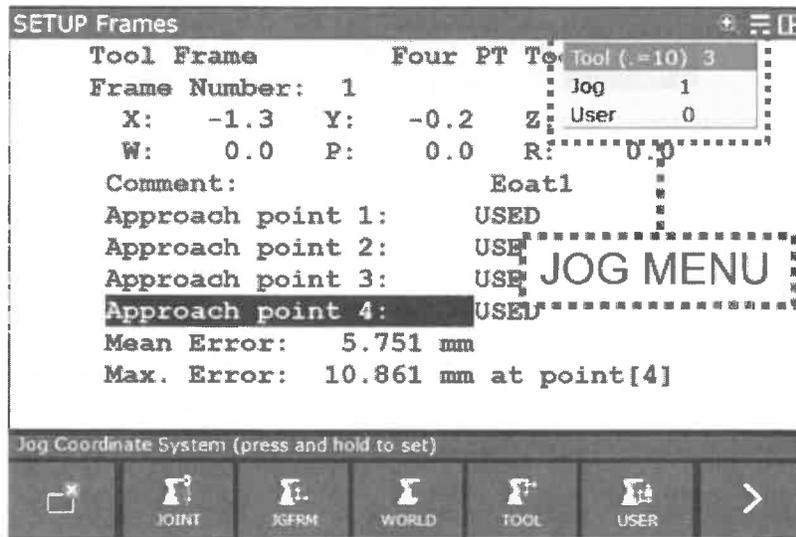
 **NOTE:** If the mean error does not decrease after re-teaching the point with max error, do not proceed any further. At this point the mean error is attributed to inaccuracies in mastering or calibration and not due to user error in teaching. If the mean error decreases after re-teaching the point with max error, it implies that the user error in teaching was corrected in the subsequent tool frame calculation.

CAUTION

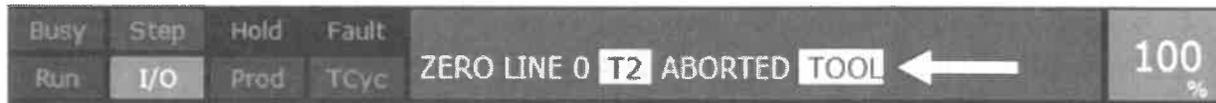
The TCP errors are calculated based on the recorded positions. It provides a guideline for the user to iteratively improve the accuracy of the recorded positions. The error measures are reasonable only when one of the four positions recorded is slightly off. When two or more recorded positions are inaccurate and deviate more from the reference position, the error feedback may not be accurate. When the difference between max and mean errors is not significantly large, more than one recorded positions deviate from the reference position.

10: Frames

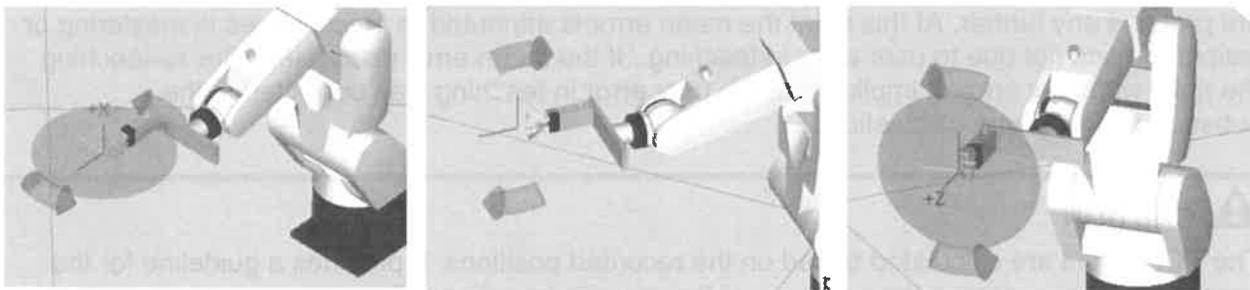
- 16 With the new Frame calculated proof the values.
 - a Measure back to the face plate from the newly defined TCP and ensure the numbers make sense.
- 17 Test the newly defined Frame by jogging as well.
 - a Hold Shift and press COORD.
 - b On the Jog Menu enter the value of the Frame you just created on the Tool line.



- c Ensure Tool is the selected Jog Method, press COORD until it is displayed.



- d Using the rotation keys, jog the robot about the X, Y, Z coordinate axes. If your frame is taught correctly the rotational movements of the robot with pivot around the point in space that you defined for your Tool Frame and not drift away from it.



10.6.3 Six Point (XY / XZ) 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 three points that define the location of the tool center point.

There are two variations of the six point method — XZ and XY. Use XZ when it is easier for you to record a point in the +z direction. Use the XY when it is easier for you to record a point in the +y direction.

Procedure 10-3 Defining a Tool Frame using the Six Point Method

- 1 Press Menu
 - a Cursor down to SETUP
 - b Press the right arrow and cursor down to Frames
 - c Press Enter
 - d You should see a screen similar to below.

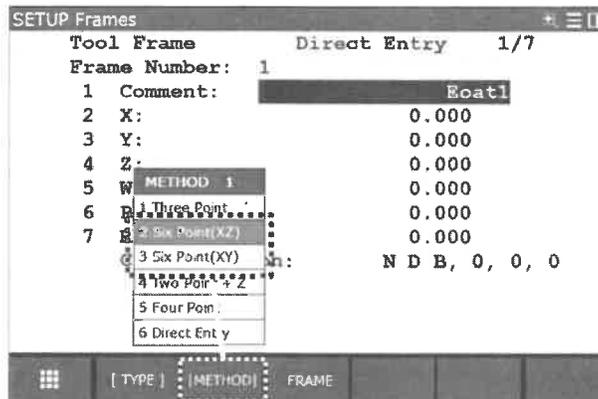
SETUP Frames					
Tool Frame	/ Direct Entry			1/10	
	X	Y	Z	Comment	
1	0.0	0.0	0.0	[Eoat1]
2	0.0	0.0	0.0	[Eoat2]
3	0.0	0.0	0.0	[Eoat3]
4	0.0	0.0	0.0	[Eoat4]
5	0.0	0.0	0.0	[Eoat5]
6	0.0	0.0	0.0	[Eoat6]
7	0.0	0.0	0.0	[Eoat7]
8	0.0	0.0	0.0	[Eoat8]
9	0.0	0.0	0.0	[Eoat9]
10	0.0	0.0	0.0	[Eoat10]

- 2 Look in the upper left hand corner, if something other than Tool Frame is displayed, press F3 [OTHER] and select Tool Frame. If OTHER is not displayed press Prev.
- 3 Place your cursor next to The Tool Frame to define and press Enter or F2 DETAIL.

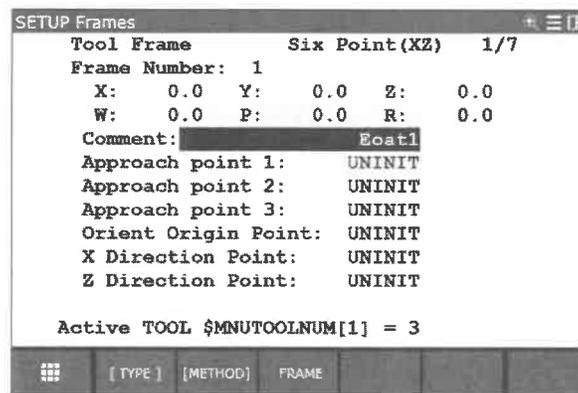
SETUP Frames		
Tool Frame	Direct Entry	1/7
Frame Number:	1	
1 Comment:	Eoat1	
2 X:	0.000	
3 Y:	0.000	
4 Z:	0.000	
5 W:	0.000	
6 P:	0.000	
7 R:	0.000	
Configuration:	N D B, 0, 0, 0	

10: Frames

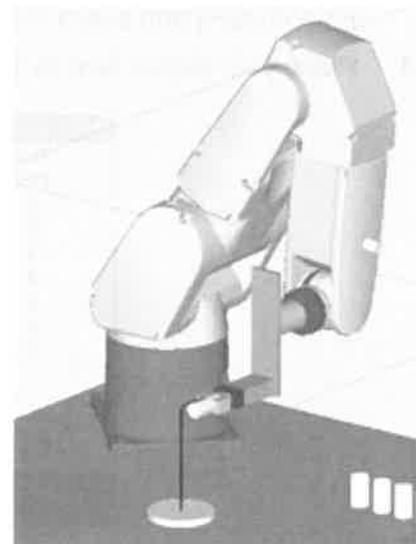
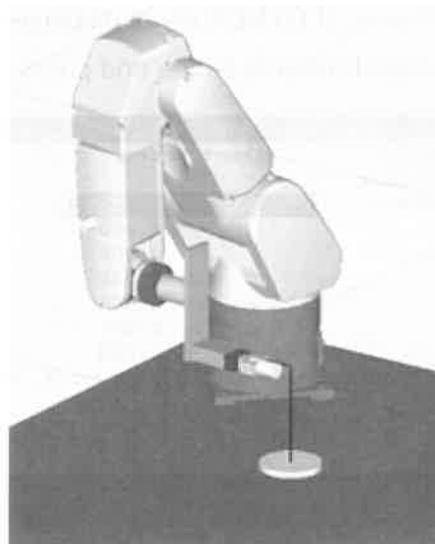
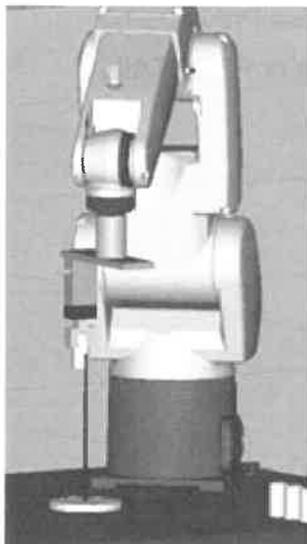
- 4 Press F2 [METHOD] highlight Six point (XY) or Six Point (XZ) and press Enter.



- 5 You should see a screen similar to below.



- 6 Add a comment to associate the Tool Frame with a specific tool.
- 7 Record Approach Points, the objective is to record the approach points from three different angles. The figures below show examples of how the approach points may be recorded



- 8 Cursor to the line labeled Approach point 1.
 - a Jog the robot so that the tool tip touches a fixed reference point
 - b With the robot properly positioned hold down Shift and F5 RECORD to record the position.
- 9 Cursor to the line labeled Approach point 2.
 - a Jog the robot so that the tool tip touches the fixed reference point but this time approach the point from a different angle.

 **NOTE:** Large rotations, near 90° but no more than 360°, give the best results. However, smaller rotations can be used if motion is restricted by cabling or other attachments.

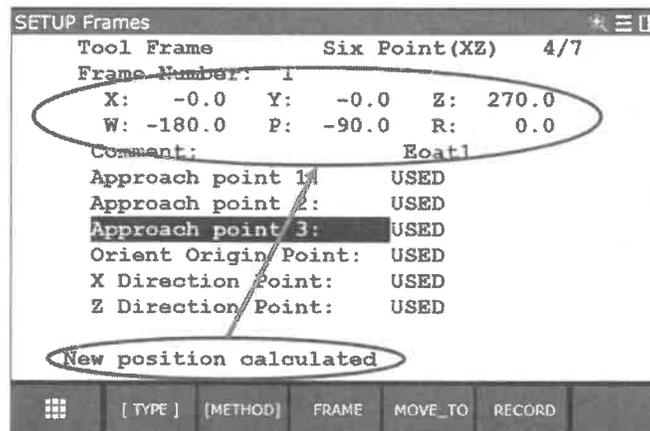
- b With the robot properly repositioned hold down the SHIFT key then press F5 RECORD.

- 10 Cursor to the line labeled Approach point 3.
 - a Jog the robot so that the tool tip touches the fixed reference point but this time approach the point from a different angle.

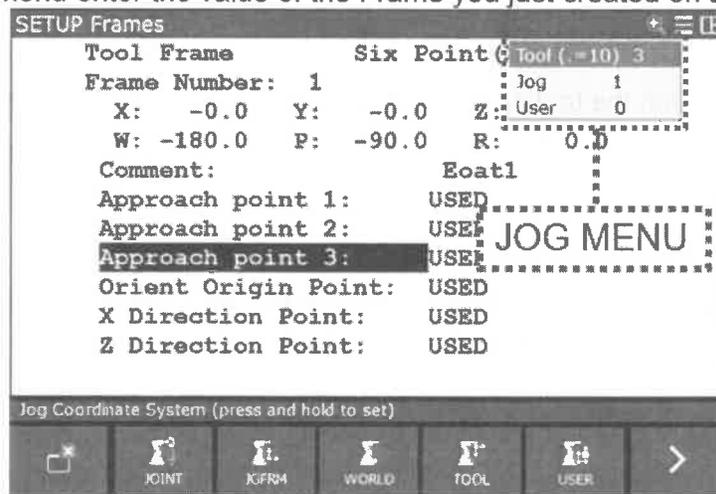
 **NOTE:** Large rotations, near 90° but no more than 360°, give the best results. However, smaller rotations can be used if motion is restricted by cabling or other attachments.

- b With the robot properly repositioned hold down the SHIFT key then press F5 RECORD.

- 11 Move the cursor to Orient Origin Point.
 - a Jog the robot so that the tooling orientation aligns with the WORLD coordinate planes.
 - b With the robot properly repositioned hold down the SHIFT key then press F5 RECORD
- 12 Move the cursor to the X Direction Point.
 - a Jog the robot in the direction that you want to set as the +X direction.
 - b With the robot properly repositioned hold down the SHIFT key then press F5 RECORD
- 13 Move the cursor to Y Direction Point or Z Direction Point (Dependent on the method selected).
 - a Jog the robot in the direction you want to set either +Y or +Z.
 - b With the robot properly repositioned hold down the SHIFT key then press F5 RECORD
- 14 After recording the final position the Frame data will be calculated. You should see a screen similar to the following page.



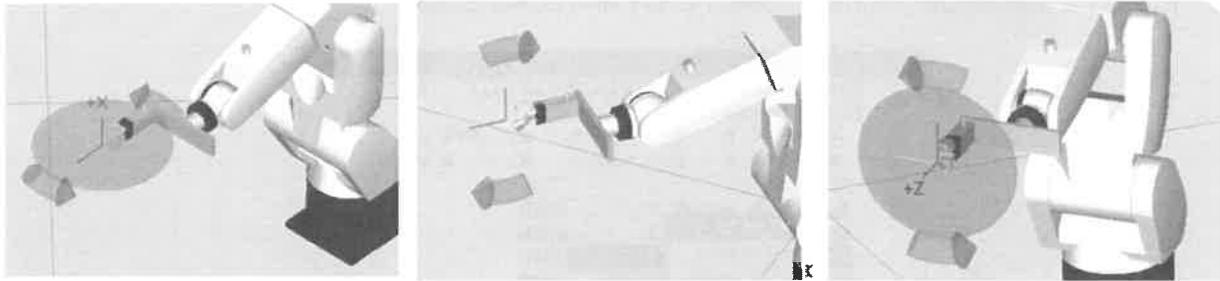
- 15 With the new Frame calculated proof the values.
 - a Measure back to the face plate from the newly defined TCP and ensure the numbers make sense.
- 16 Test the newly defined Frame by jogging as well.
 - a Hold Shift and press COORD.
 - b On the Jog Menu enter the value of the Frame you just created on the Tool line.



- 17 Ensure Tool is the selected Jog Method, press COORD until it is displayed.



- 18 Using the rotation keys, jog the robot about the X, Y, Z coordinate axes. If your frame is taught correctly the rotational movements of the robot with pivot around the point in space that you defined for your Tool Frame and not drift away from it.



19 Using the X, Y, Z keys, test the new orientation of the Tool Frame. Ensure the robot moves in the directions that you taught with the frame.

WARNING

When using Tool Frame Offsets with programs, the offset direction is relative to the orientation of the Tool Frame. Ensure you remember how the orientation of the Tool Frames so you apply proper offsets. Failure to comply may result in injury to personnel or damage to equipment.

10.6.4 Direct Entry Method

The direct entry method provides for direct recording and numerical entry of the frame position. For TCP dimensions, refer to the manufacturing specifications of the tool.

Procedure 10-4 Defining a Tool Frame using the Direct Entry Method

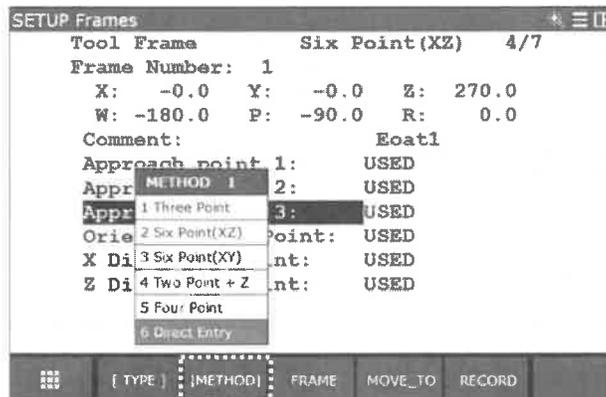
- 1 Press Menu
 - a Cursor down to SETUP
 - b Press the right arrow and cursor down to Frames
 - c Press Enter
 - d You should see a screen similar to below.

SETUP Frames					
Tool Frame	/ Direct Entry			1/10	
X	Y	Z	Comment		
1	0.0	0.0	0.0	[Eoat1]
2	0.0	0.0	0.0	[Eoat2]
3	0.0	0.0	0.0	[Eoat3]
4	0.0	0.0	0.0	[Eoat4]
5	0.0	0.0	0.0	[Eoat5]
6	0.0	0.0	0.0	[Eoat6]
7	0.0	0.0	0.0	[Eoat7]
8	0.0	0.0	0.0	[Eoat8]
9	0.0	0.0	0.0	[Eoat9]
10	0.0	0.0	0.0	[Eoat10]

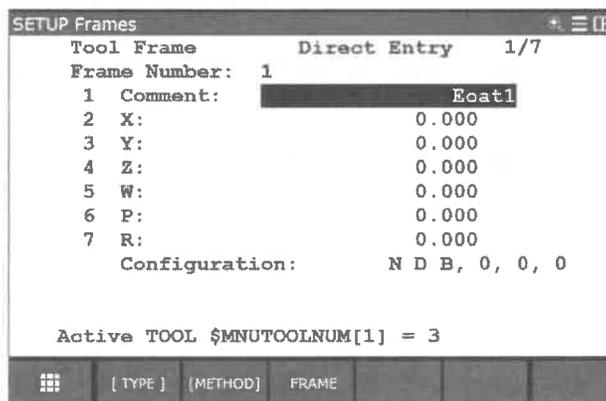
- 2 Look in the upper left hand corner, if something other than Tool Frame is displayed, press F3 [OTHER] and select Tool Frame. If OTHER is not displayed press Prev.
- 3 Place your cursor next to The Tool Frame to define and press Enter or F2 DETAIL.

10: Frames

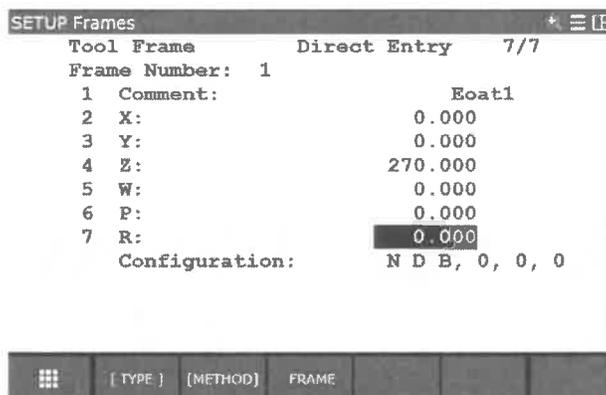
- Press F2 [METHOD] highlight Direct Entry and press Enter.



- You should see a screen similar to below.

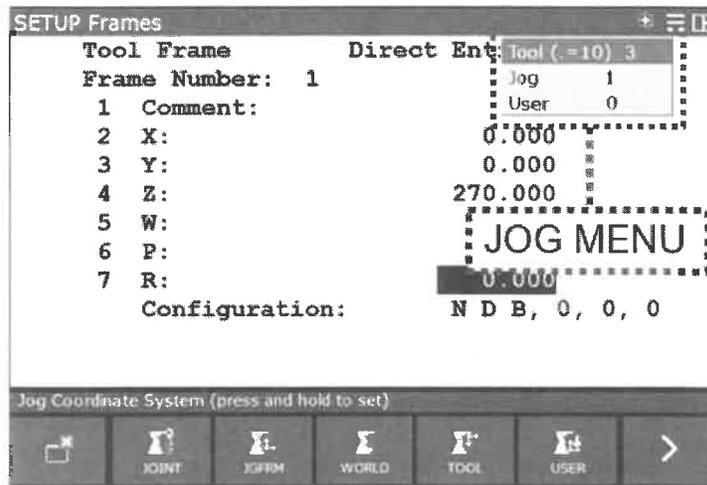


- Add a comment to associate the Tool Frame with a specific tool.
- Set each position component:
 - Move the cursor to the component.
 - Enter the numeric value for the component.
 - Press the ENTER key to set the new value



Handling Tool

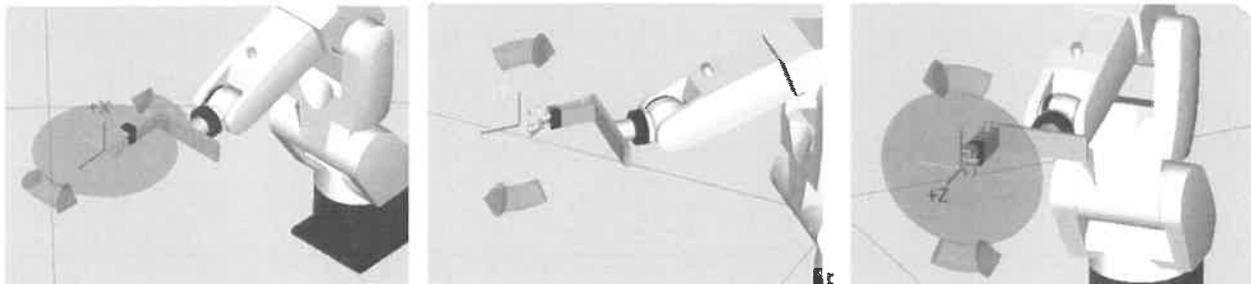
- 8 With the new Frame calculated proof the values.
 - a Measure back to the face plate from the newly defined TCP and ensure the numbers make sense.
- 9 Test the newly defined Frame by jogging as well.
 - a Hold Shift and press COORD.
 - b On the Jog Menu enter the value of the Frame you just created on the Tool line.



- 10 Ensure Tool is the selected Jog Method, press COORD until it is displayed.



- 11 Using the rotation keys, jog the robot about the X, Y, Z coordinate axes. If your frame is taught correctly the rotational movements of the robot with pivot around the point in space that you defined for your Tool Frame and not drift away from it.



- 12 If you changed the orientation of the frame, ensure you test the directions using the X, Y, and Z jog keys. The robot should respond by moving the newly defined directions.

Lab 11 Creating Tool Frames

Student Name: _____

Assignment: The student will:

- Create a Tool Frame Using Direct Entry
- Create a Three, Four, or Six Point Tool Frame

Condition: Given a FANUC Robot and a controller

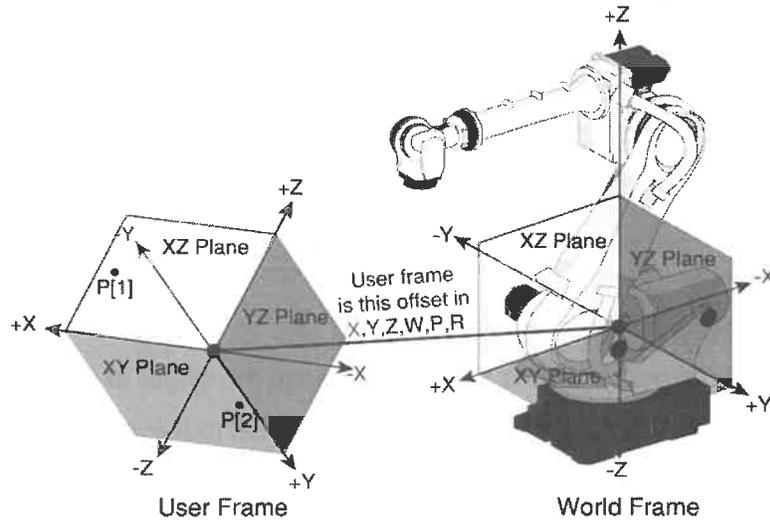
- Step:**
- 1 Create a Tool Frame using Direct Entry. (See Section 10.6.4 Direct Entry Method, pg. 131.)
 - 2 Create a Tool Frame using either
 - Section 10.6.1 Three Point Method, pg. 118,
 - Section 10.6.2 Four Point Method, pg. 121
 - Section 10.6.3 Six Point (XY / XZ) Method, pg. 127
 - 3 Test each Frame to ensure accuracy.

Completed:

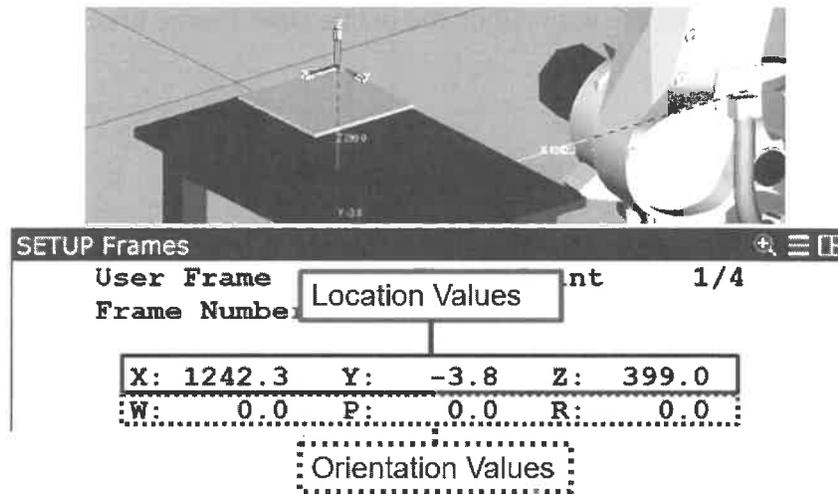
Instructor: _____

10.7 User Frames

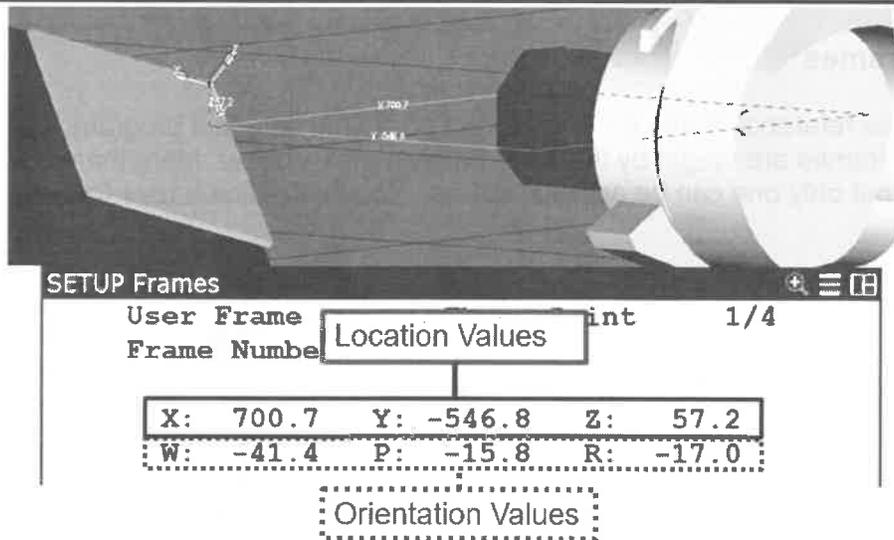
User frame is the reference frame for all recorded positional data in a program. As the name suggests, user frames are taught by the robot programmer, or user. More than one user frame can be taught, but only one can be active at a time. You can define a user frame is any location and orientation.



If you do not set up the location and orientation of the user frame before you create a program, the user frame will use World frame's location and orientation



The User Frame data is the physical distance that the origin of the frame is from the origin of the robot relative to X, Y, and Z. The W, P, and R values of the user frame identify the orientation of the frame. From the image above we can see the User Frame maintained the default orientation which is X+ forward, Y+ to the left, and Z+ up. As seen below, if we were to add a positive or negative value in the W, P, and/or R fields of the frame the orientation would have rotated about the specified plane(s).



When working with User Frames keep in mind:

- When no data is defined the User Frame location is the origin of World Frame.
- The User Frame data is a physical measurement back to the origin of the robot.
- The controller allows you to set up many different user frames for each robot, default of nine schedules.
 - R30iB Controllers can have up to 61 User Frames
 - R30iB Plus Controllers can have up to 253 User Frames
- When selecting User as the jog method the active User Frame orientation defines how the robot will jog.

10.8 Creating a User Frame

There are three standard ways to define a User Frame:

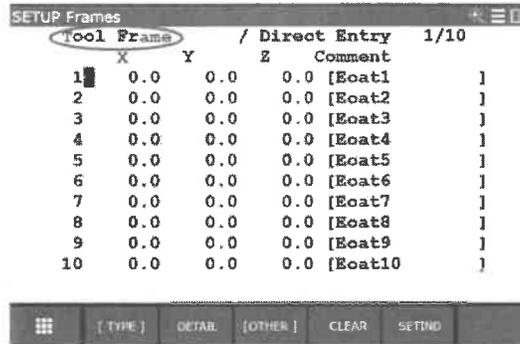
- Three Point Recording three points defines the user frame. See **10.8.1 Three Point Method**
- Four Point Use the four point method when you need to define a frame that has its origin at a position other than the reference of the frame. See **10.8.2 Four Point Method**
- Direct Entry Use the direct entry method when you know the coordinates of the user frame. See **10.8.3 Direct Entry Method**

10.8.1 Three Point Method

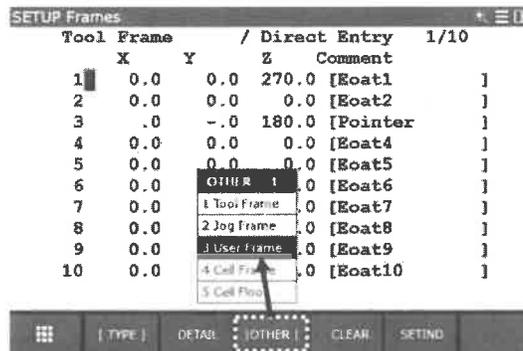
Recording three points defines the user frame. The three points are the origin, a position along the +X-axis of the user frame, and a position on the X-Y plane of the user frame (defines the X-Y plane and the Y-Z plane).

Procedure 10-5 Defining a User Frame using the Three Point Method

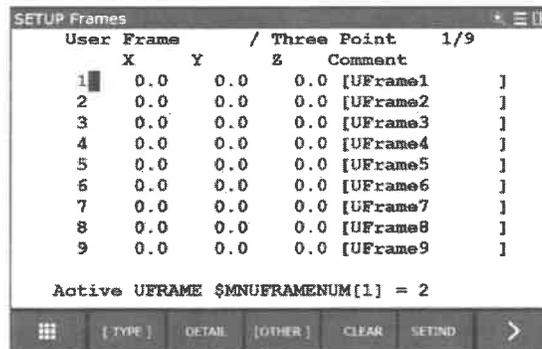
- 1 Press Menu
 - a Cursor down to SETUP
 - b Press the right arrow and cursor down to Frames
 - c Press Enter
 - d You should see a screen similar to below.



- 2 Look in the upper left hand corner, if something other than User Frame is displayed, press F3 [OTHER] and select User Frame. If OTHER is not displayed press Prev.



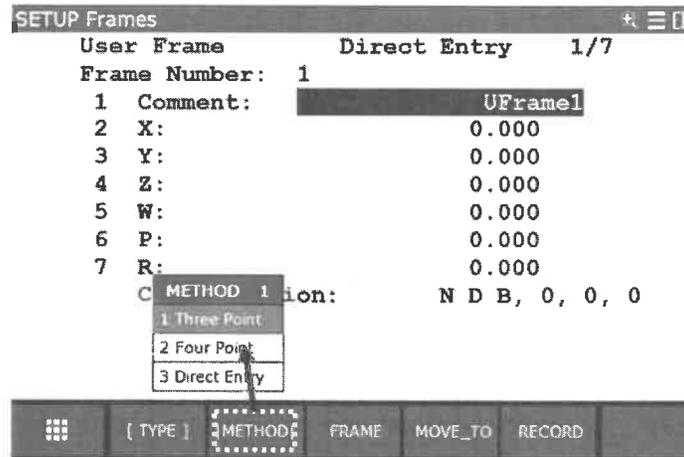
- 3 You should see a screen similar to below.



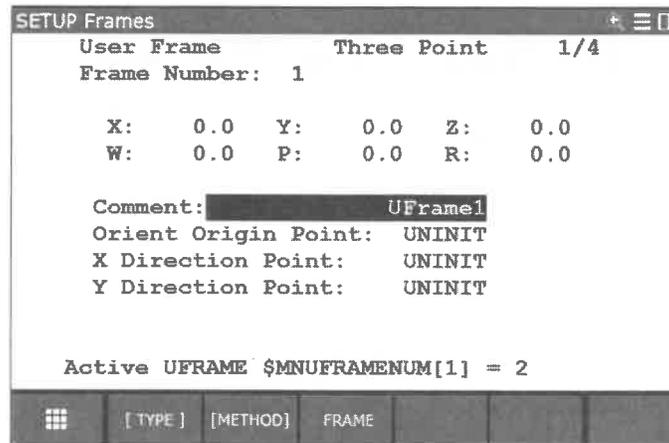
- 4 Place your cursor next to the User Frame to define and press Enter or F2 DETAIL

10: Frames

- 5 Within the frame setup screen press F2 [METHOD] and select Three Point from the list.



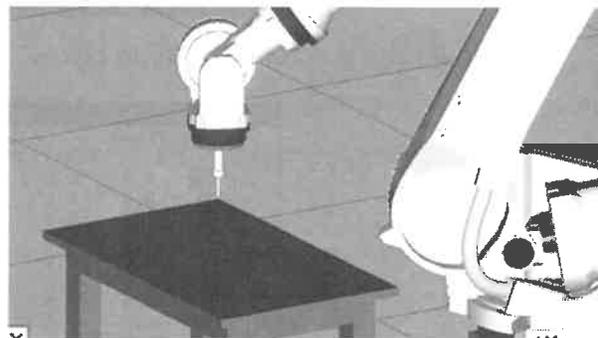
- 6 You will see a screen similar to below.



- 7 Add a comment that describes the use of the frame.

- 8 Cursor to the line labeled Orient Origin Point.

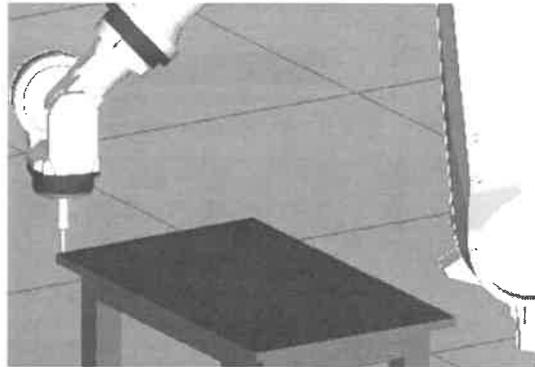
- a Jog the robot so that the tool tip touches the point to be set as the origin of the frame.
- b With the robot properly positioned hold down Shift and F5 RECORD to record the position.



NOTE: This point defines the origin of the frame relative to the origin of World Frame.

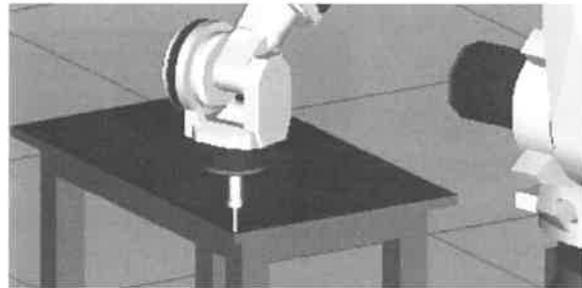
9 Cursor to the line labeled X Direction Point.

- a Jog the robot so that the tool tip touches the point to define the X Direction of the frame.
- b With the robot properly positioned hold down Shift and F5 RECORD to record the position.



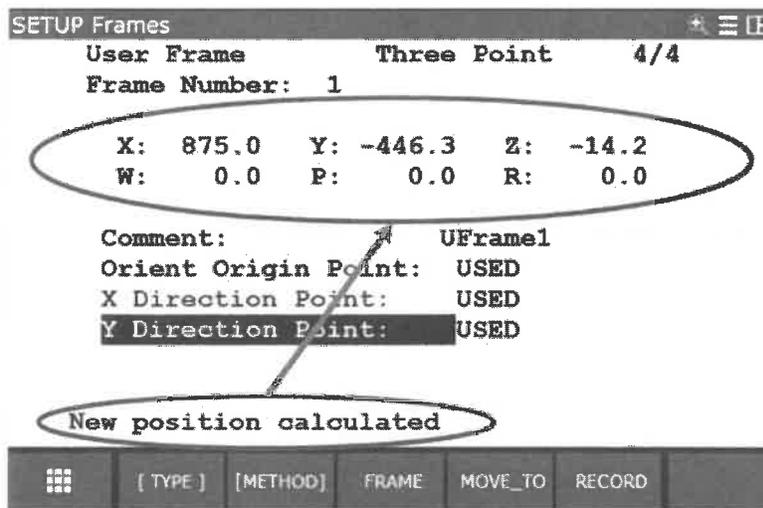
10 Cursor to the line labeled Y Direction Point.

- a Jog the robot so that the tool tip touches the point to define the Y Direction of the frame.
- b With the robot properly positioned hold down Shift and F5 RECORD to record the position.



NOTE: The X and Y directions establish the orientation of the User Frame.

11 After recording the final position the Frame data will be calculated. You should see a screen similar to below.



12 With the new Frame calculated proof the values.

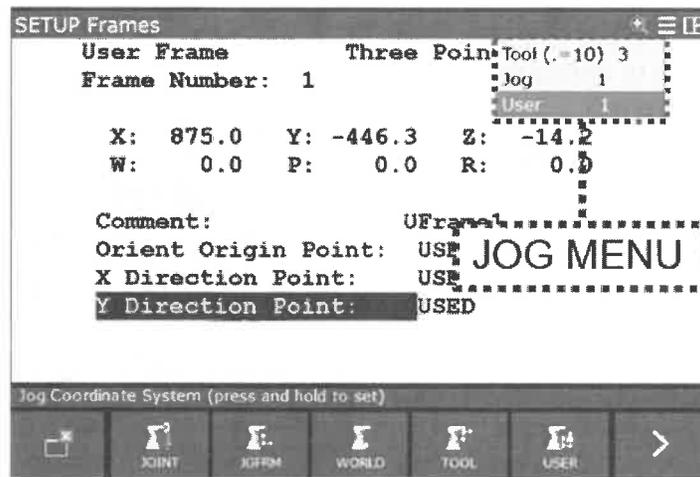
- a Measure from the origin of the User Frame back to the origin of World Frame and ensure the numbers make sense.

13 Test the newly defined Frame by jogging as well.

- a Hold Shift and press COORD.

10: Frames

- b On the Jog Menu enter the value of the Frame you just created on the User line.

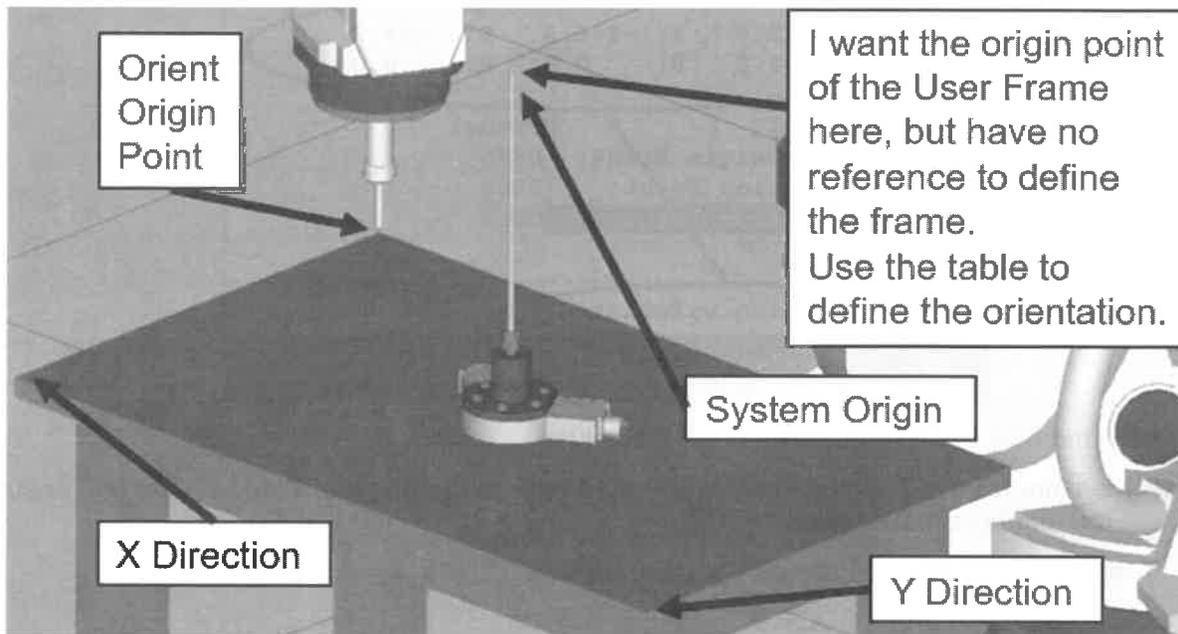


- c Ensure User is the selected Jog Method, press COORD until it is displayed.



10.8.2 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. You can also use it 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).



Procedure 10-6 Defining a User Frame using the Four Point Method

- 1 Press Menu
 - a Cursor down to SETUP
 - b Press the right arrow and cursor down to Frames
 - c Press Enter
 - d You should see a screen similar to below.

Tool Frame	X	Y	Z	Comment	
1	0.0	0.0	0.0	[Eoat1	1
2	0.0	0.0	0.0	[Eoat2	1
3	0.0	0.0	0.0	[Eoat3	1
4	0.0	0.0	0.0	[Eoat4	1
5	0.0	0.0	0.0	[Eoat5	1
6	0.0	0.0	0.0	[Eoat6	1
7	0.0	0.0	0.0	[Eoat7	1
8	0.0	0.0	0.0	[Eoat8	1
9	0.0	0.0	0.0	[Eoat9	1
10	0.0	0.0	0.0	[Eoat10	1

- 2 Look in the upper left hand corner, if something other than User Frame is displayed, press F3 [OTHER] and select User Frame. If OTHER is not displayed press Prev.

Tool Frame	X	Y	Z	Comment	
1	0.0	0.0	270.0	[Eoat1	1
2	0.0	0.0	0.0	[Eoat2	1
3	.0	-.0	180.0	[Pointer	1
4	0.0	0.0	0.0	[Eoat4	1
5	0.0	0.0	0.0	[Eoat5	1
6	0.0	0.0	0.0	[Eoat6	1
7	0.0	0.0	0.0	[Eoat7	1
8	0.0	0.0	0.0	[Eoat8	1
9	0.0	0.0	0.0	[Eoat9	1
10	0.0	0.0	0.0	[Eoat10	1

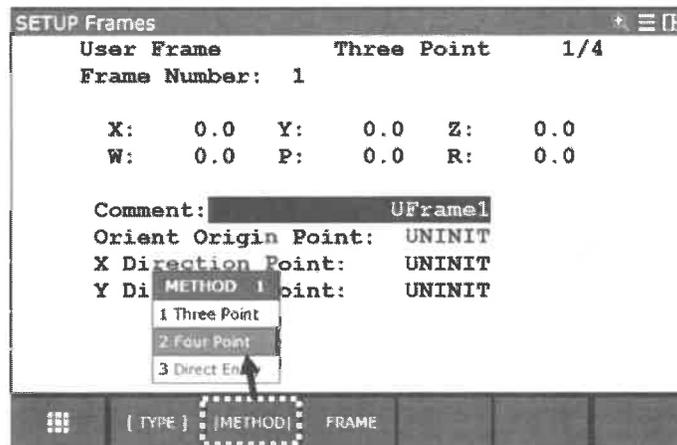
- 3 You should see a screen similar to below.

User Frame	X	Y	Z	Comment	
1	0.0	0.0	0.0	[UFrame1	1
2	0.0	0.0	0.0	[UFrame2	1
3	0.0	0.0	0.0	[UFrame3	1
4	0.0	0.0	0.0	[UFrame4	1
5	0.0	0.0	0.0	[UFrame5	1
6	0.0	0.0	0.0	[UFrame6	1
7	0.0	0.0	0.0	[UFrame7	1
8	0.0	0.0	0.0	[UFrame8	1
9	0.0	0.0	0.0	[UFrame9	1

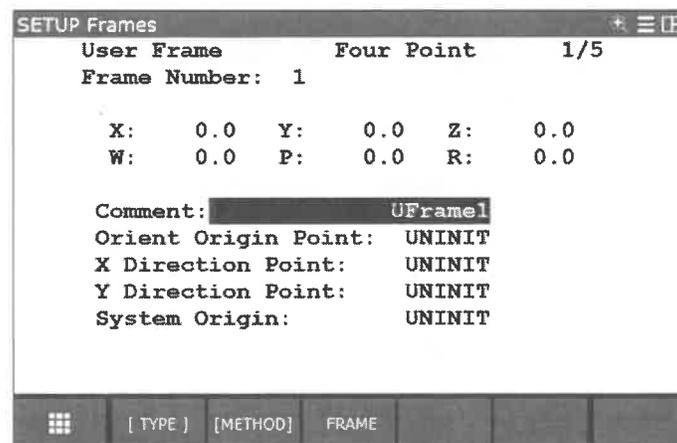
Active UFRAME \$MNUFRAMENUM[1] = 2

10: Frames

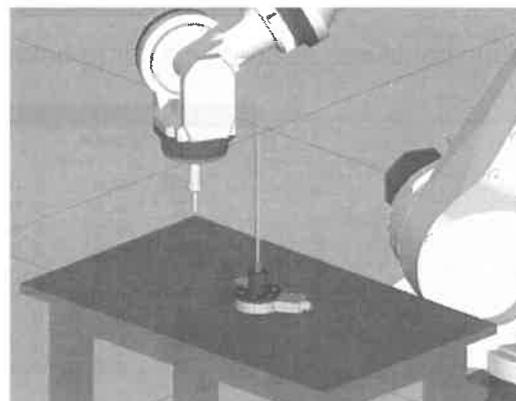
- 4 Place your cursor next to the User Frame to define and press Enter or F2 DETAIL
- 5 Within the frame setup screen press F2 [METHOD] and select Four Point from the list.



- 6 You will see a screen similar to below.

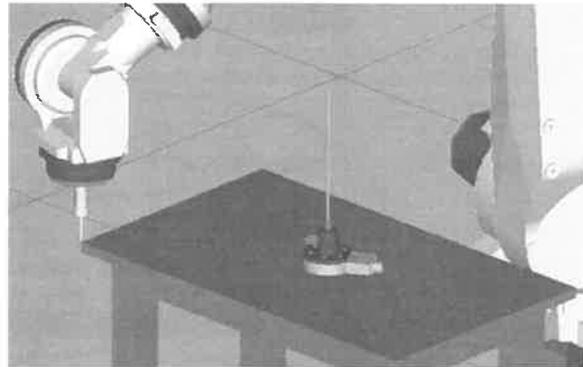


- 7 Add a comment that describes the use of the frame.
- 8 Cursor to the line labeled Orient Origin Point.
 - a Jog the robot so that the tool tip touches the point to be set as the origin of the frame.
 - b With the robot properly positioned hold down Shift and F5 RECORD to record the position.

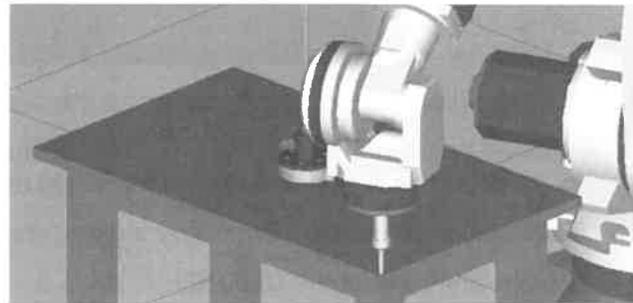


NOTE: This point defines the origin for the **orientation** of the frame.

- 9 Cursor to the line labeled X Direction Point.
- Jog the robot so that the tool tip touches the point to define the X Direction of the frame.
 - With the robot properly positioned hold down Shift and F5 RECORD to record the position.

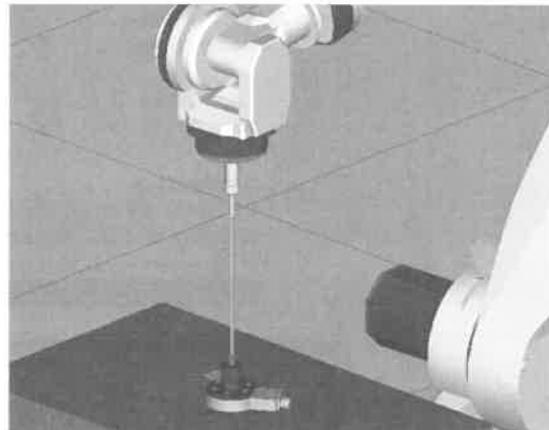


- 10 Cursor to the line labeled Y Direction Point.
- Jog the robot so that the tool tip touches the point to define the Y Direction of the frame.
 - With the robot properly positioned hold down Shift and F5 RECORD to record the position.



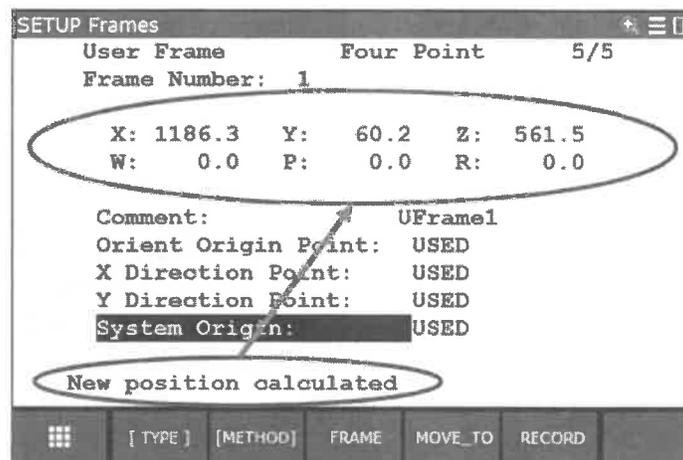
 **NOTE:** The X and Y directions establish the orientation of the User Frame.

- 11 Cursor to the line labeled System Origin Point.
- Jog the robot so that the tool tip touches the point to define the Y Direction of the frame.
 - With the robot properly positioned hold down Shift and F5 RECORD to record the position.

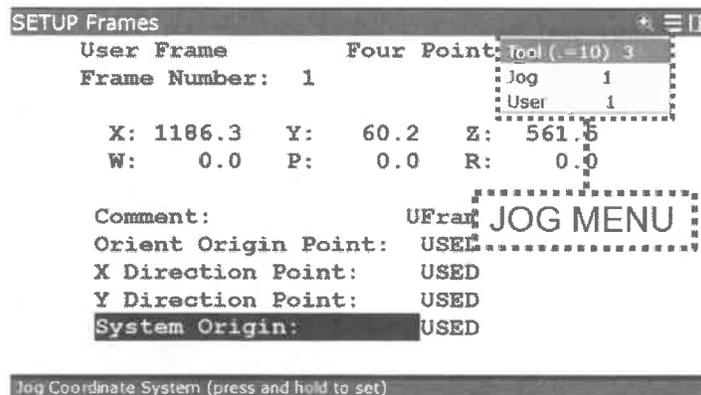


 **NOTE:** The system origin point establishes the origin point for the user frame. This is the position that all data will be measured from using this frame. This will also establish the distance of the frame relative to the world origin.

- 12 After recording the final position the Frame data will be calculated. You should see a screen similar to below.



- 13 With the new Frame calculated proof the values.
 - a Measure from the origin of the User Frame (system origin point) back to the origin of World Frame and ensure the numbers make sense.
- 14 Test the newly defined Frame by jogging as well.
 - a Hold Shift and press COORD.
 - b On the Jog Menu enter the value of the Frame you just created on the User line.



- c Ensure User is the selected Jog Method, press COORD until it is displayed.



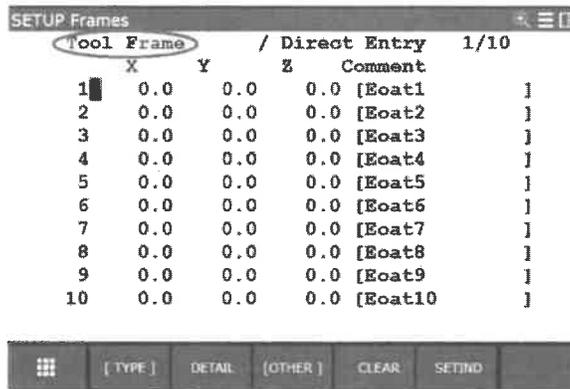
10.8.3 Direct Entry Method

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.

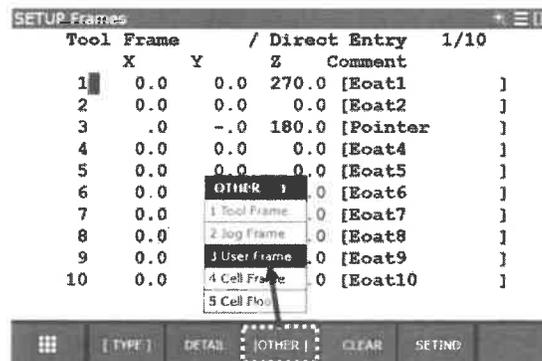
Procedure 10-7 Define a User Frame using the Direct Entry Method

- 1 Press Menu

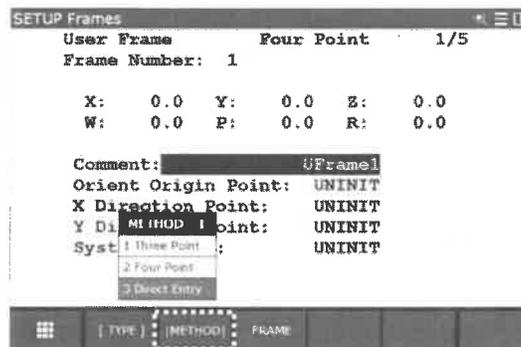
- a Cursor down to SETUP
- b Press the right arrow and cursor down to Frames
- c Press Enter
- d You should see a screen similar to below.



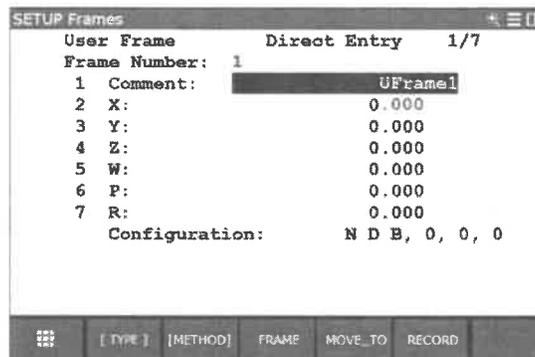
- 2 Look in the upper left hand corner, if something other than User Frame is displayed, press F3 [OTHER] and select User Frame. If OTHER is not displayed press Prev.



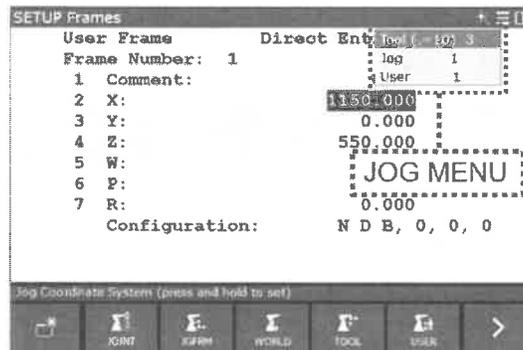
- 3 Place your cursor next to the User Frame to define and press Enter or F2 DETAIL
- 4 Within the frame setup screen press F2 [METHOD] and select Direct Entry from the list.



- 5 You should see a screen similar to below.



- 6 Add a comment to associate the Tool Frame with a specific tool.
- 7 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
- 8 With the new Frame calculated proof the values.
 - a Measure from the origin of the User Frame (system origin point) back to the origin of World Frame and ensure the numbers make sense.
- 9 Test the newly defined Frame by jogging as well.
 - a Hold Shift and press COORD.
 - b On the Jog Menu enter the value of the Frame you just created on the User line.

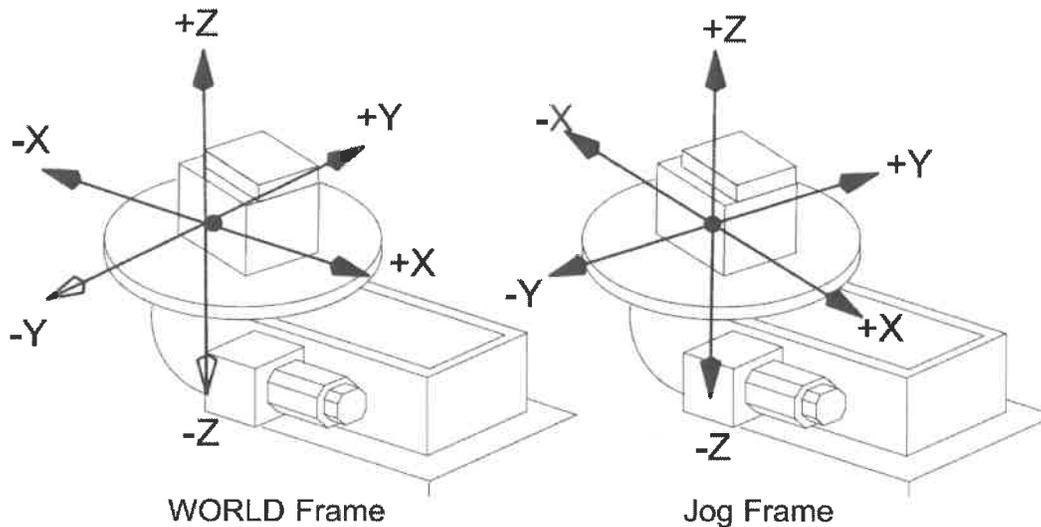


- c Ensure User is the selected Jog Method, press COORD until it is displayed.



10.9 Jog Frames

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.



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.

When working with Jog Frames keep in mind:

- 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 per robot group.
- You can jog the robot in jog frame.
- Positional data is **not** dependent on a Jog Frame.

10.10 Creating Jog Frames

There are two standard ways to define a Jog Frame:

- **Three Point** Recording three points defines the user frame. See **10.8.1 Three Point Method**
- **Direct Entry** Use the direct entry method when you know the coordinates of the user frame. See **10.8.3 Direct Entry Method**

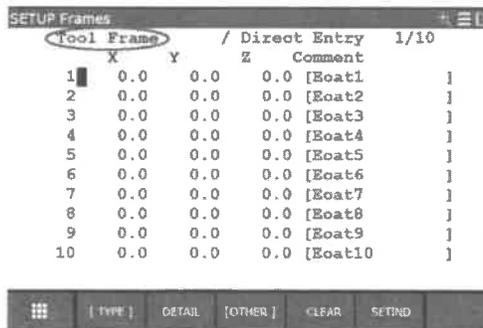
10: Frames

10.10.1 Three Point Method

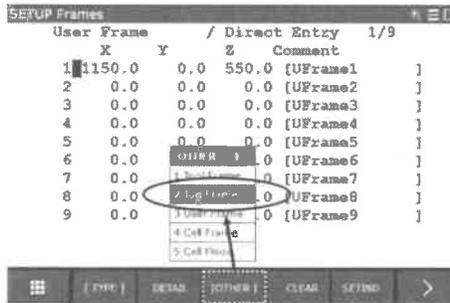
The three point method allows you to define a jog frame by recording three points: the origin, a point along the +x-axis of the user frame, and a point on the x-y plane of the user frame (defines the x-y plane and the y-z plane).

Procedure 10-8 Defining a Jog Frame using the Three Point Method

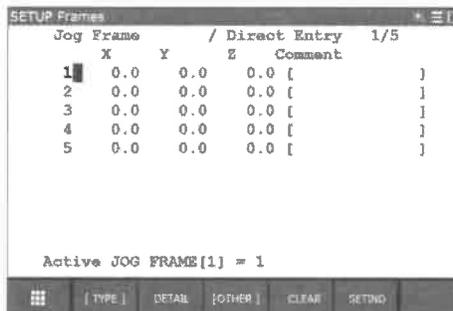
- 1 Press Menu
 - a Cursor down to SETUP
 - b Press the right arrow and cursor down to Frames
 - c Press Enter
 - d You should see a screen similar to below.



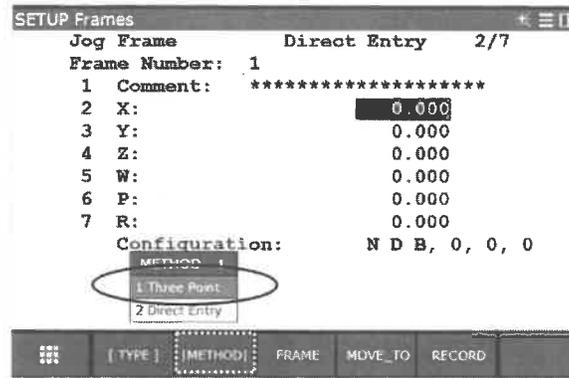
- 2 Look in the upper left hand corner, if something other than Jog Frame is displayed, press F3 [OTHER] and select Jog Frame. If OTHER is not displayed press Prev.



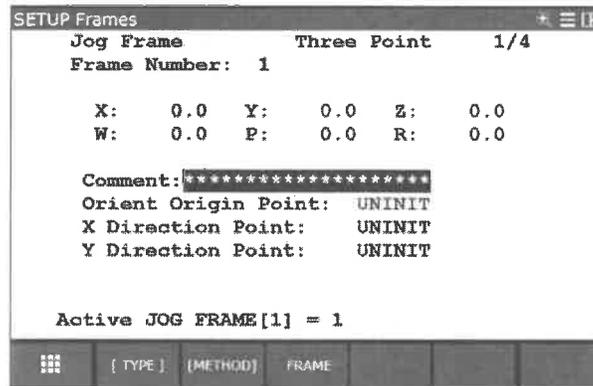
- 3 You should see a screen similar to below.



- 4 Place your cursor next to the Jog Frame to define and press Enter or F2 DETAIL
- 5 Within the frame setup screen press F2 [METHOD] and select Three Point from the list.

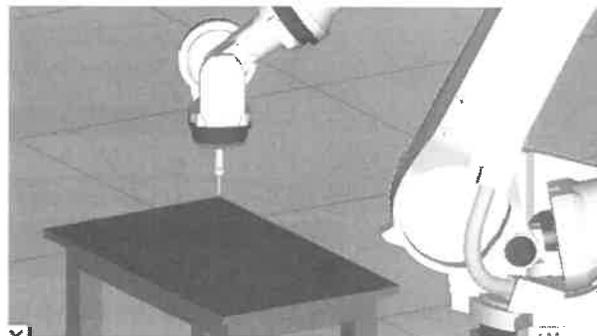


- 6 You will see a screen similar to below.



- 7 Add a comment that describes the use of the frame.
- 8 Cursor to the line labeled Orient Origin Point.

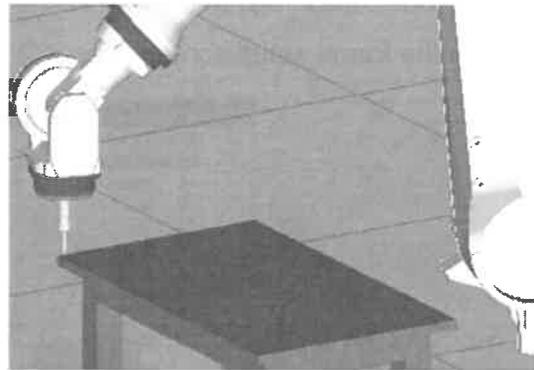
- a Jog the robot so that the tool tip touches the point to be set as the origin of the frame.
- b With the robot properly positioned hold down Shift and F5 RECORD to record the position.



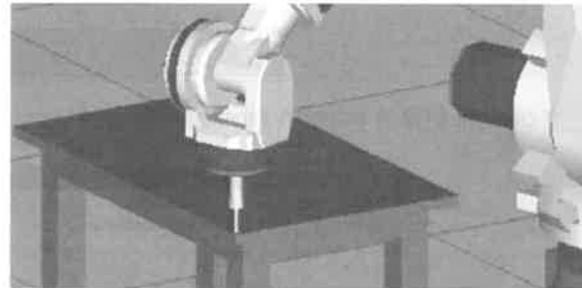
NOTE: This point defines the origin of the frame relative to the origin of World Frame.

10: Frames

- 9 Cursor to the line labeled X Direction Point.
 - a Jog the robot so that the tool tip touches the point to define the X Direction of the frame.
 - b With the robot properly positioned hold down Shift and F5 RECORD to record the position.

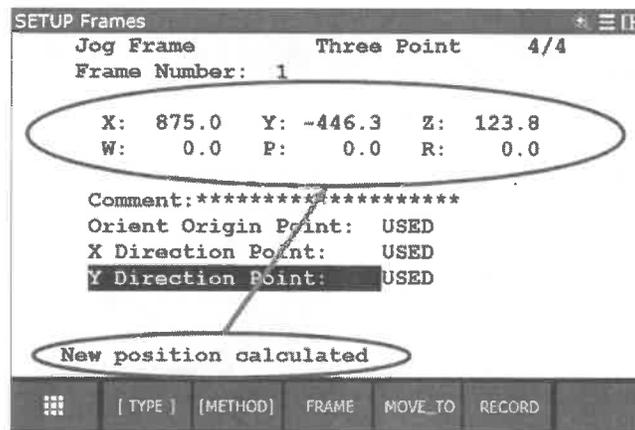


- 10 Cursor to the line labeled Y Direction Point.
 - a Jog the robot so that the tool tip touches the point to define the Y Direction of the frame.
 - b With the robot properly positioned hold down Shift and F5 RECORD to record the position.

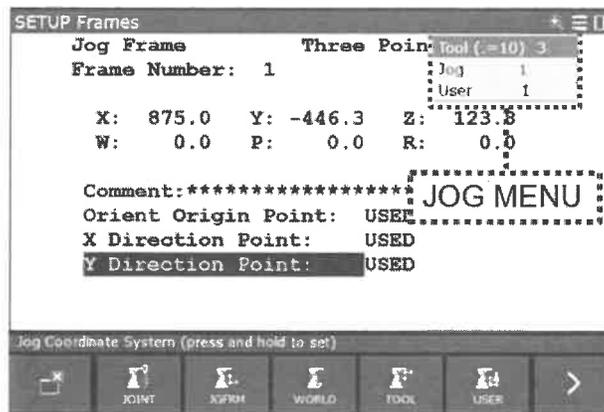


NOTE: The X and Y directions establish the orientation of the User Frame.

- 11 After recording the final position the Frame data will be calculated. You should see a screen similar to below.



- 12 With the new Frame calculated proof the values.
 - a Measure from the origin of the Jog Frame back to the origin of World Frame and ensure the numbers make sense.
- 13 Test the Newly defined Frame by jogging as well.
 - a Hold Shift and press COORD.
 - b On the Jog Menu enter the value of the Frame you just created on the Jog line.



c Ensure JGFRM is the selected Jog Method, press COORD until it is displayed.

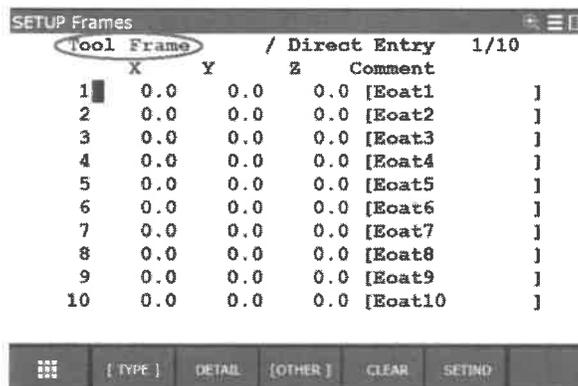


10.10.2 Direct Entry Method

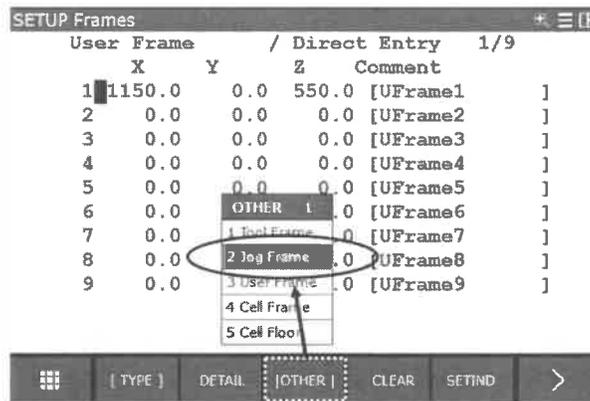
The direct entry method allows you to designate the origin with values for X, Y, Z, W, P, and R. This method provides direct recording and numerical entry of the frame position.

Procedure 10-9 Defining the Jog Frame using the Direct Entry Method

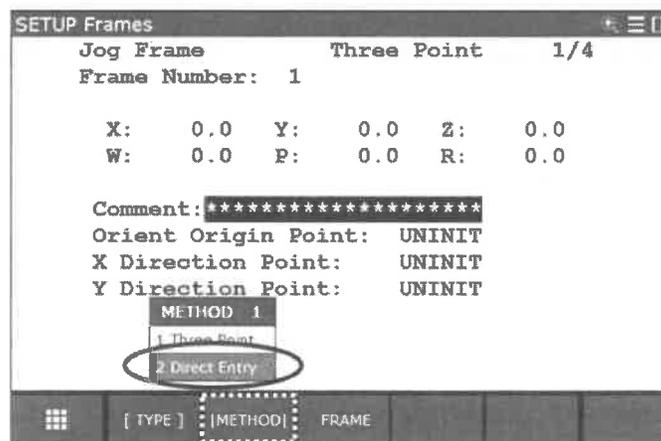
- 1 Press Menu
 - a Cursor down to SETUP
 - b Press the right arrow and cursor down to Frames
 - c Press Enter
 - d You should see a screen similar to below.



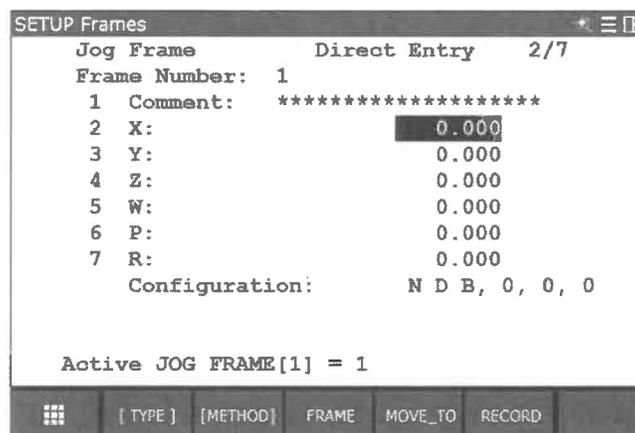
- 2 Look in the upper left hand corner, if something other than Jog Frame is displayed, press F3 [OTHER] and select Jog Frame. If OTHER is not displayed press Prev.



- 3 Place your cursor next to the Jog Frame to define and press Enter or F2 DETAIL
- 4 Within the frame setup screen press F2 [METHOD] and select Direct Entry from the list.



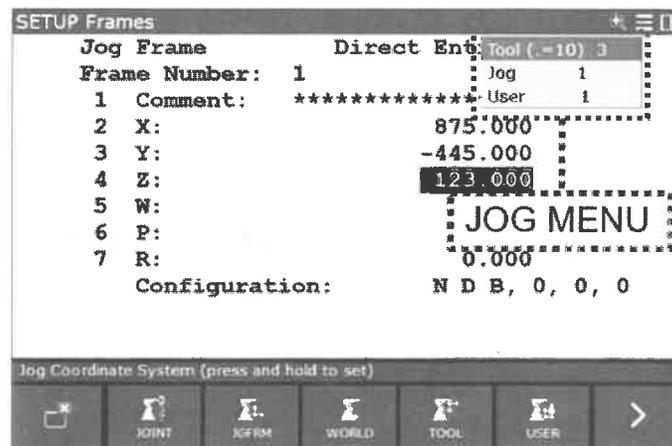
- 5 You should see a screen similar to below.



- 6 Add a comment to associate the Jog Frame with a specific tool.
- 7 Set each position component:
 - a Move the cursor to the component.

HandlingTool

- b Enter the numeric value for the component.
- c Press the ENTER key to set the new value
- 8 With the new Frame calculated proof the values.
 - a Measure from the origin of the Jog Frame (system origin point) back to the origin of World Frame and ensure the numbers make sense.
- 9 Test the newly defined Frame by jogging as well.
 - a Hold Shift and press COORD.
 - b On the Jog Menu enter the value of the Frame you just created on the User line.



- c Ensure JGFRM is the selected Jog Method, press COORD until it is displayed.



10.11 Saving and Restoring Frame Data

Once Frames have been created the data will automatically be saved to internal controller memory (SRAM). It is a good idea to ensure this data is also saved to the default device. Having this information as a backup will be handy in case the data on the controller is changed, become corrupt, etc.

Frame data can be saved as a part of a complete file backup (7.5 File Backup) or just the individual files that store frame information. The data can also be restored as part of complete file restore (7.6 File Restore) or restored via the individual files.

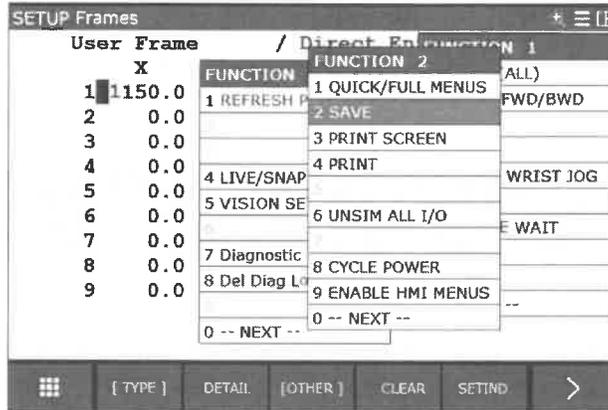
The files that make up your frame data are:

- **FRAMEVAR.VR**: Jog Frame data and comments for all frames.
- **SYSFRAME.SV**: Tool Frame and User Frame data.

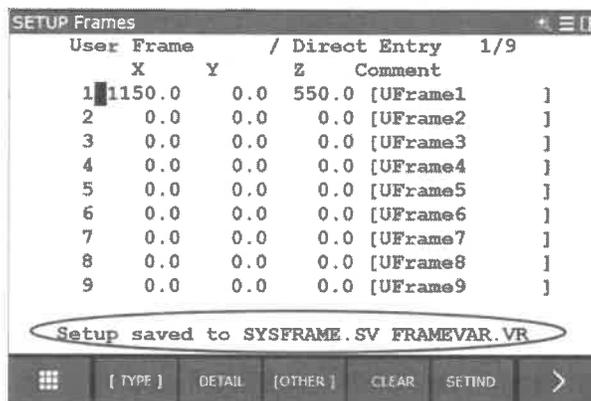
NOTE: This sections contains steps to save and restore just the frame data only. If you require a complete file backup or restore refer to **Chapter 7**.

Procedure 10-10 Saving Frame Data

- 1 Press Menu.
- 2 Cursor down to File.
 - a Press the right arrow.
 - b Select File from the submenu
- 3 Set the default device, see **7.4.1 Setting the Default Device**.
- 4 Navigate to the appropriate subdirectories (if necessary), see **7.4.3 Subdirectories**.
- 5 Display any frame screen.
- 6 Press FCTN.
- 7 Select SAVE.



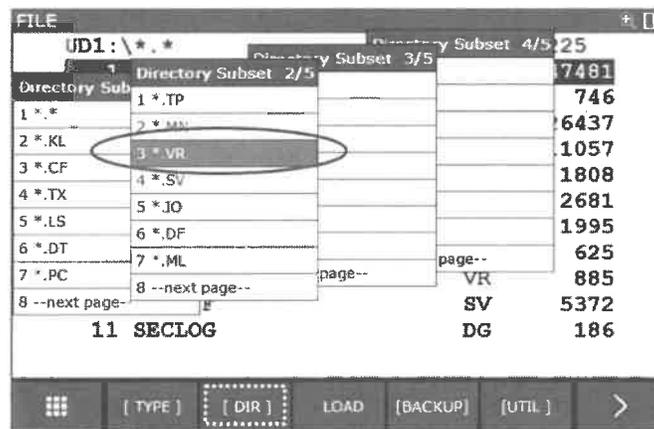
- 8 Both of the files, FRAMEVAR.VR and SYSFRAME.SV will be saved to the default device.



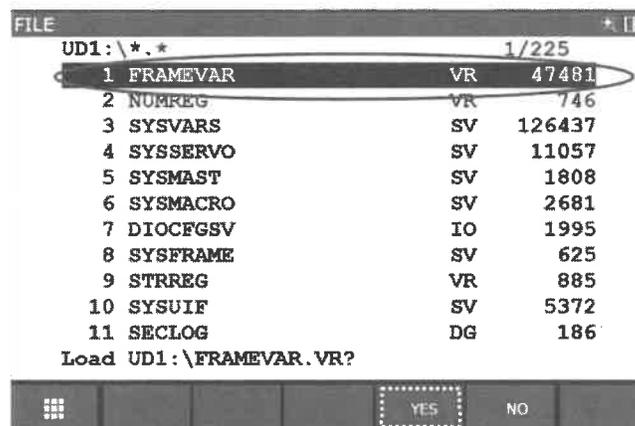
10.11.2 Restoring Frame Data

Procedure 10-11 Restoring Frame Data

- 1 Press Menu.
- 2 Cursor down to File.
 - a Press the right arrow.
 - b Select File from the submenu.
- 3 Set the default device, see 7.4.1 **Setting the Default Device**.
- 4 Navigate to the appropriate subdirectories (if necessary), see 7.4.3 **Subdirectories**.
- 5 Press F2 DIR and select .VR.



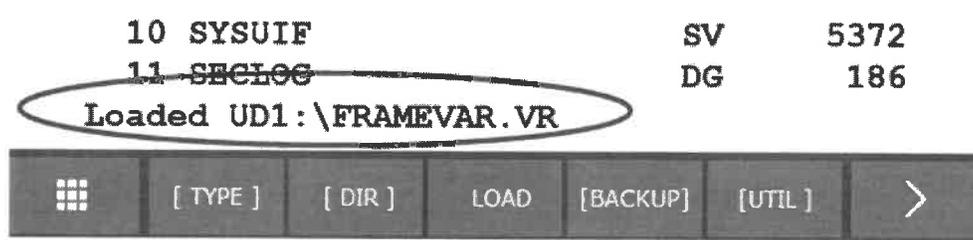
- 6 Highlight the file named FRAMEVAR.VR.
- 7 Press F3 LOAD.
- 8 Press F4 YES to verify you want to load the file.



- 9 You will see a prompt indicating that the file has been loaded.

10: Frames

HandlingTool



10 Follow Steps 5-9 to load the SYSFRAME.SV file.

 **NOTE:** You must load both the FRAMEVAR.VR and the SYSFRAME.SV files to recover the frame data and frame comments.

10.12 Chapter Review**1. Where is the default Tool Center Point?**

- A Intersection of J1 and J2
- B Center of the Faceplate
- C Tip of the default tool
- D FANUC robots do not use Tool Center Points

2. What two buttons display the Jog Menu

- A Shift and Display
- B Menu and any jog key
- C Shift and COORD
- D Group and COORD

3. The six point method of defining a Tool Frame defines the Location and _____ of a Tool Frame.

- A Distance
- B Orientation
- C Operation
- D Offset

4. Positional data is dependent on what two frames?

- A World and User
- B Jog and Tool
- C User and Tool
- D Tool and World

Lab 12 User Frames

Student Name: _____

Assignment: The student will:

- Create a User Frame Using Direct Entry.
 - Create a Three or Four Point User Frame.
-

Condition: Given a FANUC Robot and a controller

- Step:**
- 1 Create a User Frame using Direct Entry. (See Section 10.8.3 Direct Entry Method, pg 144).
 - 2 Create a User Frame using either
 - Section 10.8.1 Three Point Method, pg. 136
 - Section 10.8.2 Four Point Method, pg. 140.
 - 3 Test each Frame to ensure accuracy.

Completed:

Instructor: _____

Lab 13
Jog Frames

Student Name: _____**Assignment:** The student will:

- Create a Jog Frame.
-

Condition: Given a FANUC Robot and a controller

- Step: 1** Create a Jog Frame using either:
- **Section 10.10.1 Three Point Method, pg. 148.**
 - **Section 10.10.2 Direct Entry Method, pg. 151.**
- 2** Test each Frame to ensure accuracy.

Completed:

Instructor: _____

Lab 14 Saving Frame Data

Student Name: _____

Assignment: The student will:

- Save Frame Specific Files
-

Condition: Given a FANUC Robot and a controller

- Step:**
- 1 Create a Directory on your Default Device named "INDIVIDUAL FILES". (See Section 7.4.1 Setting the Default Device, pg. 71 and Section 7.4.3 Subdirectories, pg. 74)
 - 2 Save the files that contain Frame Data. (See Section 10.11.1 Saving Frame Data, pg. 154)

Completed:

Instructor: _____

11 CREATING A PROGRAM

11.1 Chapter Objectives

- Understand what a motion program is
- Select Screen Overview
- Edit Screen Overview
- Create a Program
- Understand Program Detail information
- Record positions in a program
- Test a Program

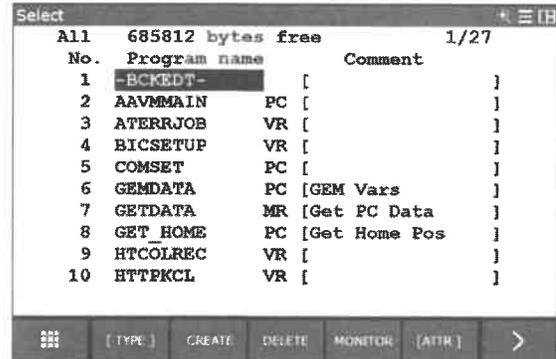
11.2 Motion Program Overview

A program is used to tell your robot what to do and how to do it. You must first create a program, and then you can add instructions to it that your robot can follow to perform a task. For example, a program can direct the robot and controller to:

- Move the robot in an appropriate way to required locations in the workcell.
- Perform an application-specific operation
- 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.

11.3 Select Screen Overview

The Select Screen is the location that all of the stored programs are located. Programs are listed in alphabetical order and can also be sorted by type. The first column displays the program name and the second column is the attribute field.



To access the Select Screen press the Select Key, you will see a screen similar to the right.

11.3.1 Select Screen Soft Keys

The Select Screen has two pages of soft keys that perform a number of different user tasks as they relate to programs. Press the Next key to toggle between soft key pages.



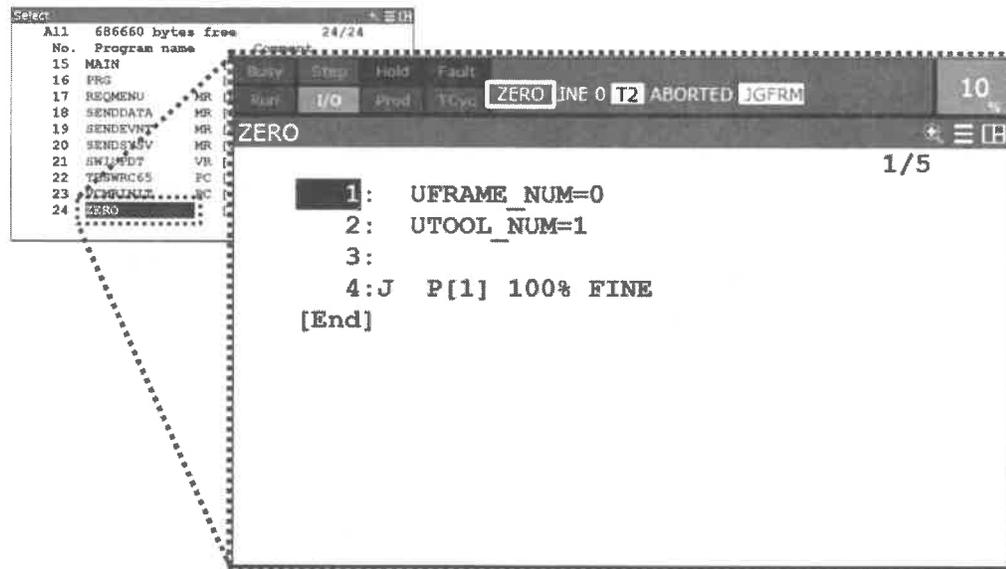
11: Creating a Program

The function of each soft key is as follows:

- [TYPE]: Allows to display programs by subtype.
- CREATE: Displays the Create Teach Pendant Program Screen to create new programs.
- DELETE: Will delete selected program.
- MONITOR: Monitors the program that is currently being executed.
- [ATTR]: Brings up the menu to select different attributes to display.
- COPY: Copies the selected program to another file name.
- DETAIL: Displays the detail of the selected program.
- LOAD: Allows loading saved programs from the default device.
- SAVE AS: Saves selected program to the default device.
- PRINT: Saves the selected program to the default device as .LS extension (readable file).

11.3.2 Selecting a Program

From the Select Screen to choose a program for execution, testing, or editing simply highlight the target program and press Enter. This will display the program in the edit screen, you will also see the program name populate the active program field along the top of the teach pendant.



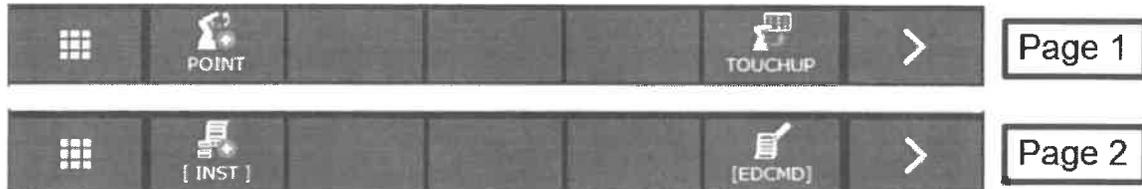
11.4 Edit Screen Overview

The Edit Screen is the location that you can display your programs for editing, testing and reviewing. As shown in the image above, the name of the selected program will be displayed along the top left hand side of the screen as well as in the top title bar of the iPendant.

- Adding instructions or editing the program is accomplished with the soft keys.

11.4.1 Edit Screen Soft Keys

The Edit Screen has two pages of soft keys that allow you to record positions, touch up positions, add program instructions, and edit. Press the Next key to toggle between soft key pages.



The function of each soft key is as follows:

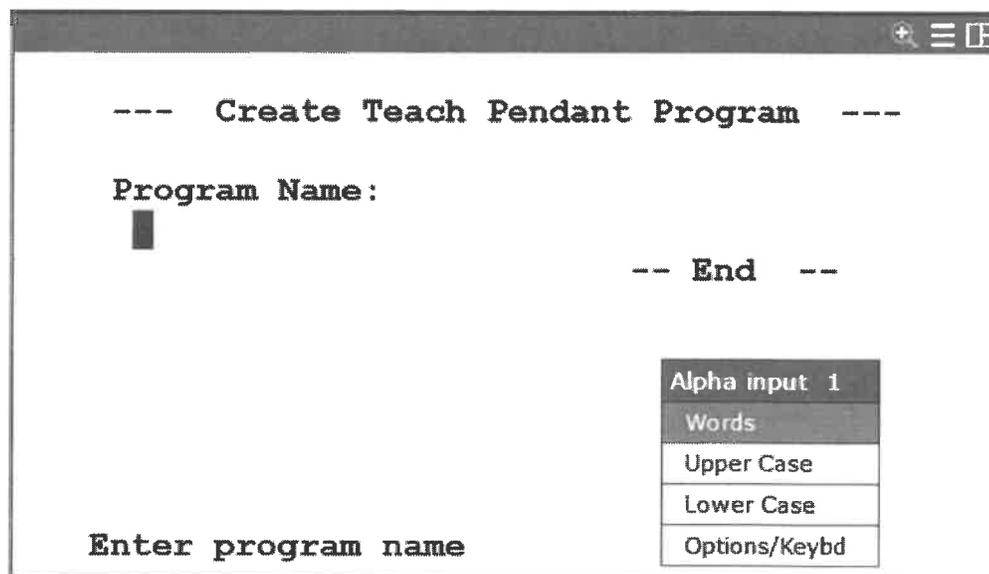
- **POINT:** Pressed with Shift, records the current position of the robot.
- **TOUCHUP:** Pressed with shift with overwrite the positional data of a previously recorded point with the current position of the robot.
- **[INST] (Instruction):** Gives access to programming instructions for logic and branching operations.
- **[EDCMD] (Edit Command):** Gives access to editing functions.

11.5 Creating a Program

Prior to viewing or editing a program, obviously it must be created. As mentioned in the previous sections, programs are created and listed on the Select screen.

Procedure 11-1 Creating a Program

- 1 Press the Select key.
- 2 Press F2, CREATE. If CREATE is not displayed press the next key.



11: Creating a Program

- 3 Enter the name of the program, the program name should be something descriptive of what the program is to do.

NOTE: The program name cannot include @ (at), * (asterisk), or spaces. You can utilize the _ (underscore); it must start with a letter, and can have a maximum of 36 characters.

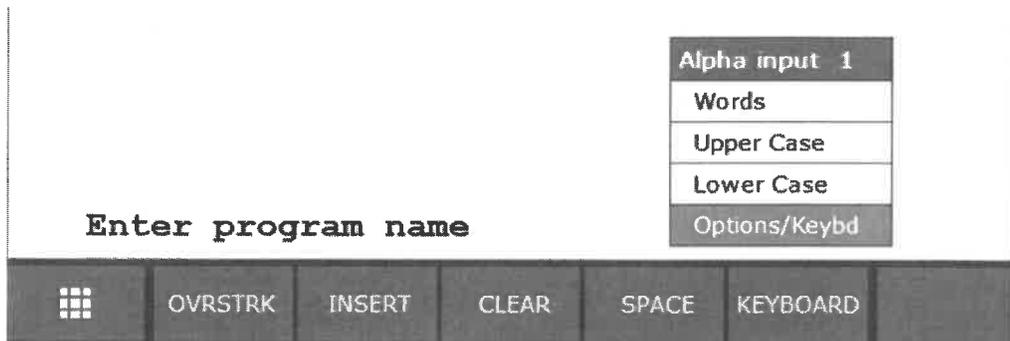
- a Utilize the preformed words, by depressing the applicable soft key.



- b Or, cursor down to Upper Case, utilize the Upper Case letters displayed on the soft keys and type the program name.



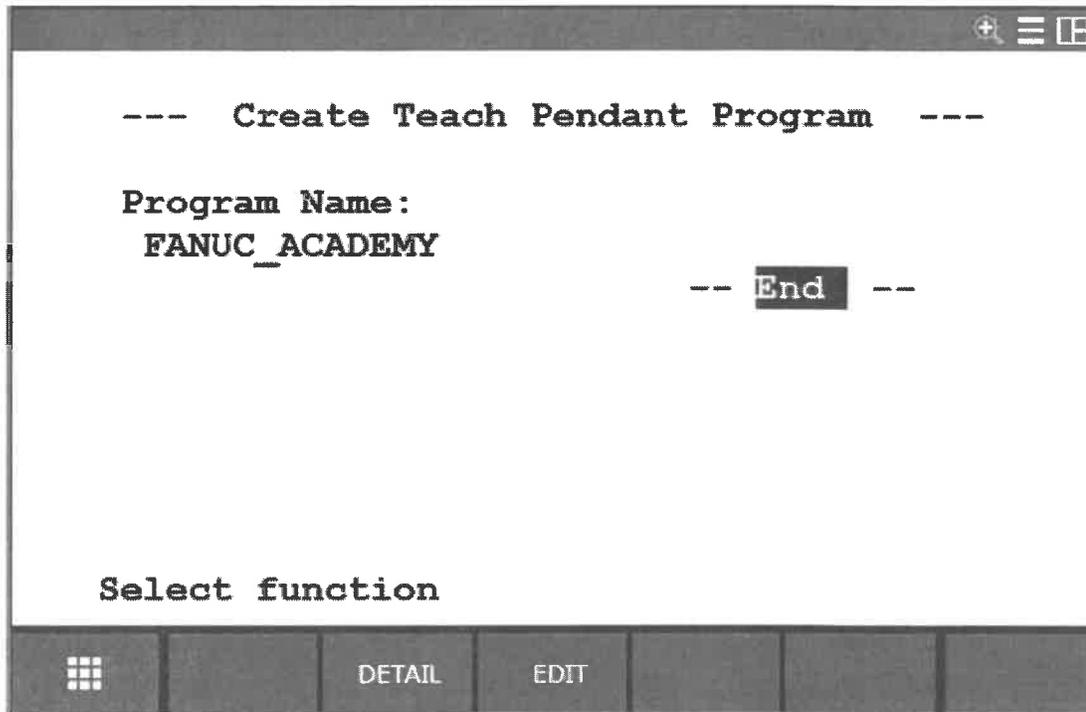
- c Or, cursor down to Options/Keybd, press F5 KEYBOARD and type in the program name.



NOTE: The program name will default to upper case after creation, so typing case sensitive is not necessary.

11: Creating a Program

- 4 Once named, press either F2 DETAIL or F3 EDIT.
 - a Pressing DETAIL will display the program detail information.
 - b Pressing EDIT will allow you to immediately begin adding instructions to the program.



11.6 Program Detail Information

Program detail information, also known as program header information, is specific information that identifies and classifies the program. Once a program has been created, the program detail information can be reviewed and/or changed.

The items that can be viewed from the program detail screen are:

- Creation Date
- Modification Date
- Copy Source
- Positions
- Program Size
- Program Name
- Sub Type
- Comment
- Group Mask
- Write Protect
- Ignore Pause
- Stack Size

Program Detail information is displayed only the first time you create a program by pressing F2 DETAIL after you have named the program.

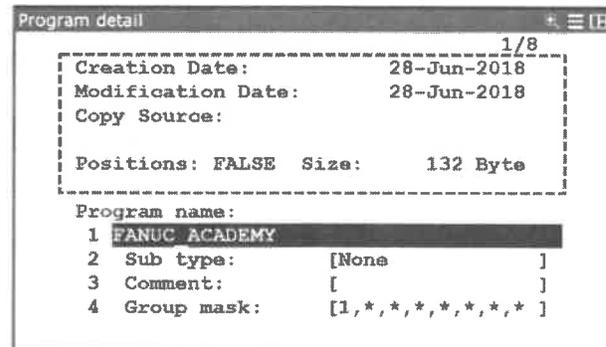
If you want to view the Program Detail information again after the program has been created, you must display it by pressing the SELECT key and then press the soft key F2 DETAIL

11: Creating a Program

11.6.1 Creation Date, Modification Date, Copy Source, Positions, and Program Size

The Creation Date, Modification Date, Copy Source, Positions, and Program Size detail information is automatically populated in the Program Detail once the program is created or modified.

- **Creation Date:** Creation date is the date on which the program name was created.
- **Modification Date*:** Modification date is the date, according to the calendar in the controller, when the file was last displayed in the editor.
- **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.
- **Positions:** The positions item indicates whether the program contains recorded robot positions. When you first create a program, positions is always set to FALSE.
- **Program Size*:** Program size is the size of the program in bytes.

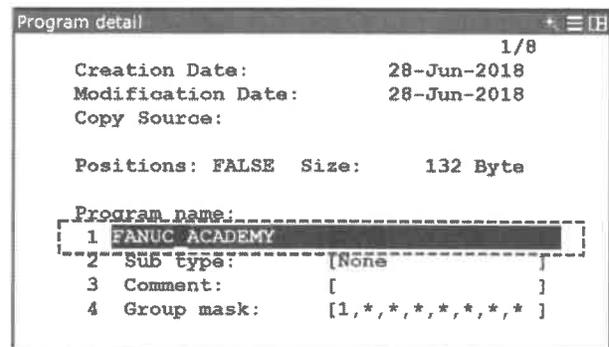


*This information can be displayed using the [ATTR] function key on the SELECT menu.

11.6.2 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 36 characters in length.
- **Characters:** All numbers and letters are allowed, however the 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.



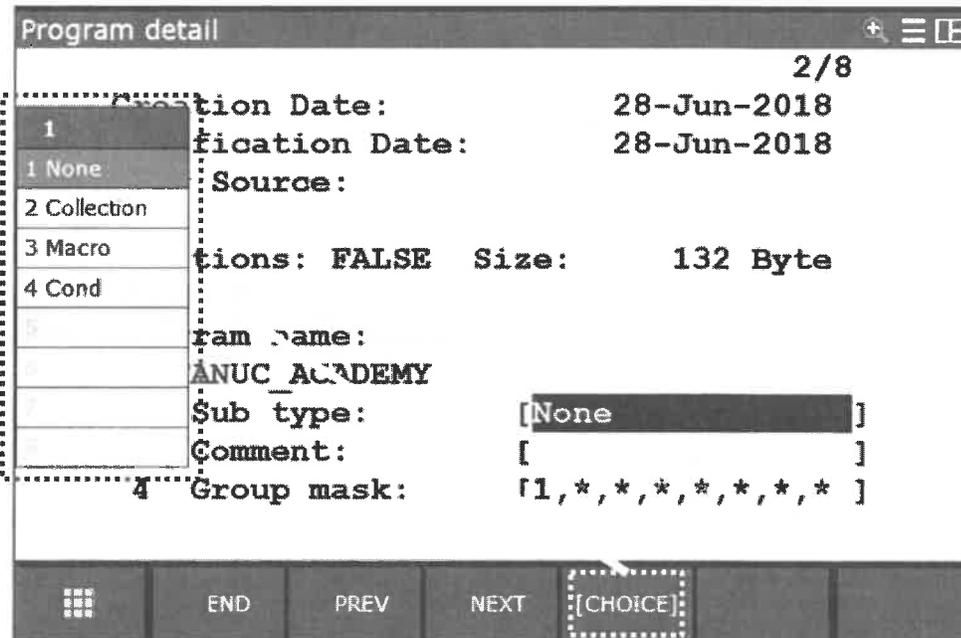
You can change the program name from the Program Detail page by placing your cursor on the existing name, press enter and rename as needed.

11: Creating a Program

11.6.3 Sub Type

Sub type identifies the kind of program you want to write. These are:

- **None:** If you select none, the program will be created as a .TP program, which can include any instructions in your teach pendant program.
- **Macro:** A macro program is created as a .MR program and can contain any instruction and function as a normal .TP program. However, only macro programs can be set up to be executed in a variety of ways including from operator panel buttons, teach pendant keys, and the Manual Functions menu. They can also be assigned a name in the macro table and be called with this name in a program.
- **Cond (Condition Handler):** A condition handler program is created as a .CH program. This type of program 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 condition handler program calls a specified teach pendant program to executed and interrupts the current program.
- **Collection:** Collections are used to organize programs together to simplify program maintenance and are created as a .CO. These are not executable programs, however they are used to organize existing programs. A good use of a collection would be to organize all of the subroutines called in a MAIN.TP program.



You can specify the sub type of program by placing your cursor of the Sub Type line, press F4 [CHOICE] and choose the appropriate sub type from the list

11: Creating a Program

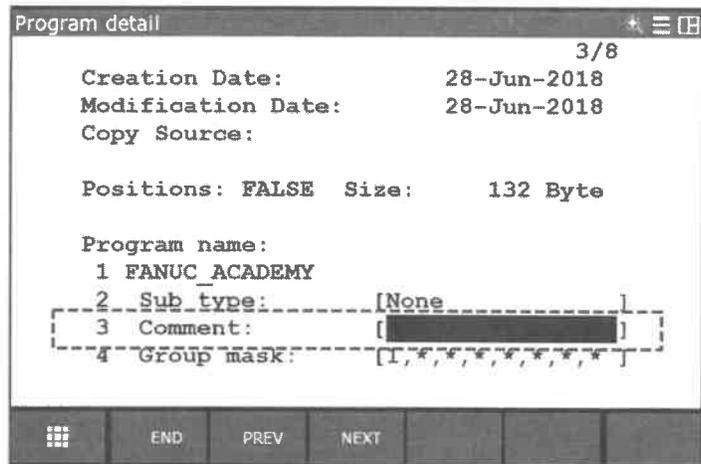
11.6.4 Program Comment

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.

To add a comment place your cursor on the comment line, press ENTER, and type in the comment.

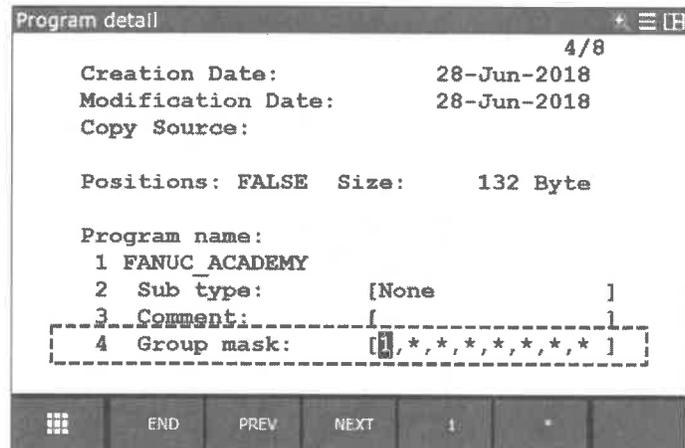
Comments must be between one to 16 characters in length. All numbers, letters, symbols, and special characters are allowed.



11.6.5 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, opening devices, and other axes.

There are eight motion groups available. The controller can have a maximum of 72 axes, however, only nine axes can belong to a single motion group.



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 single program to use all eight motion groups, but only two motion groups can perform Cartesian interpolated motion within that program.

If you disable all groups, you cannot add motion instructions to your program. You cannot change the group mask after you have added motion instructions to your program. You will not be able to select group mask in the program DETAIL screen.

11.6.6 Write Protection

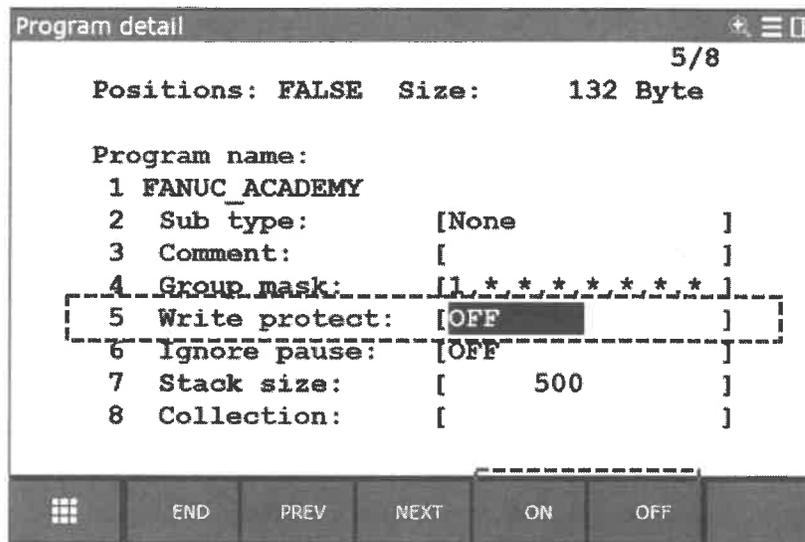
Write protection allows you to specify whether the program can be modified.

- 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 inadvertently modify the program.*

11: Creating a Program

- 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.

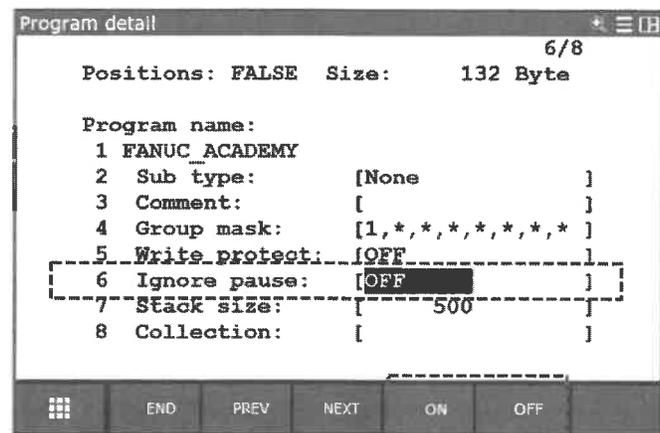
NOTE: This information can be displayed using the [ATTR] function key on the SELECT menu.



11.6.7 Ignore Pause

If the program type has been set to NONE, you can use ignore pause 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 only works in programs that do not have motion groups specified, such as an I/O monitoring program. This can be set in any program, however you will only see the effects in a non-motion program.**

- 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.



11: Creating a Program

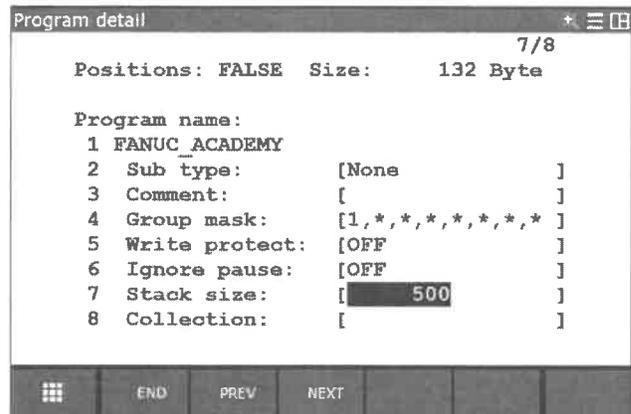
11.6.8 Stack Size

When a program is executed, a stack of 500 long words (2000 bytes) is allocated unless you specify a stack size. The stack is allocated from available user RAM.

If a stack overflow alarm occurs, the program needs more stack. Options such as TCP Mate and Cell Finder require that teach pendant programs have increased stack size.

The minimum and default stack size is 500. The maximum stack size is 4000.

To increase or decrease stack size highlight the value and enter the new amount using the numeric keypad.



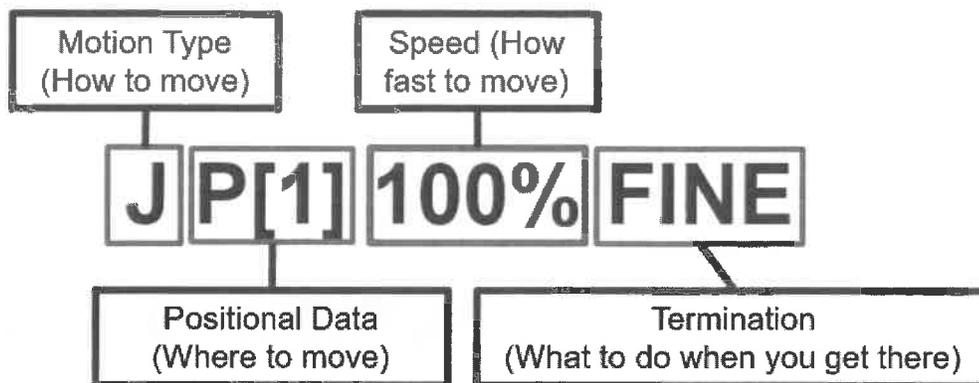
11.7 Adding Motion Instructions to a Program

Once a program has been created, instructions must be added to tell the robot how to perform a task. Instructions that tell the robot where to go, how to move there, at what speed to move, and what to do when you get there are called motion instructions.

A motion instruction includes:

- Motion Type – How the robot moves to the position.
- Positional Data – Where the robot moves.
- Speed – How fast the robot moves to a position.
- Termination Type – How the robot ends the move to the position.

The image below is an example of a motion instruction:



11: Creating a Program

11.7.1 Motion Instructions and Frames

As discussed in the Frames Chapter, Frames are used to describe the location and orientation of a position in three-dimensional space. When you record positions in a program via a Motion Instruction, this positional data is relative to the active Tool Frame and User Frame.

- **Tool Frame Data:** Tells the controller where the Tool Center point is relative to the center of the faceplate
- **User Frame Data:** Tells the controller where the origin of the User Frame is relative to the origin of the robot.
- **Positional Data:** Tells the controller where the Tool Frame is relative to the User Frame.

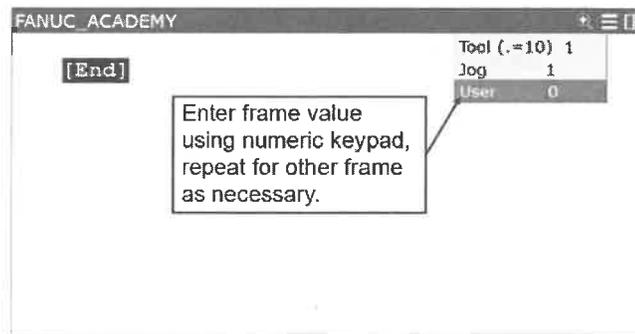
You must set the active Tool and User Frame that will be used when teaching points. You can manually set the frames using the Jog Menu or via a program instruction. **A program taught in one User Frame and Tool Frame combination will not run if a different User Frame and Tool Frame combination is active.**

Procedure 11-2 Manually Setting the User Frame and Tool Frame

- 1 Press and hold SHIFT and COORD.



- 2 On the jog menu enter the correct value for the Tool Frame or User Frame on the applicable line.



- 3 Repeat for the other frame as necessary.

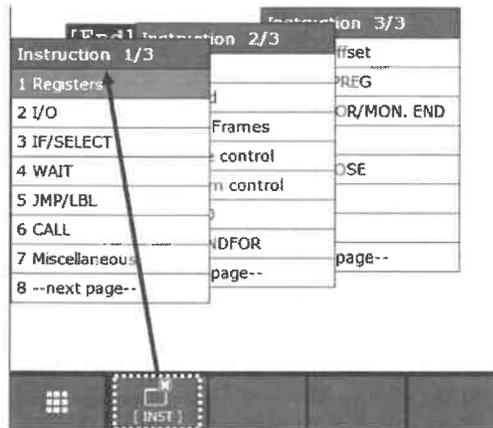
11: Creating a Program

Procedure 11-3 Setting the User Frame and Tool Frame as a Program Instruction

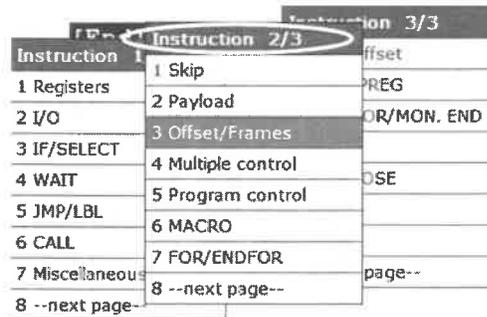
- 1 Place the cursor on the line you want to add the instruction.

 **NOTE:** If the cursor is on a line with instructions already they will be overwritten when you add the frame instruction.

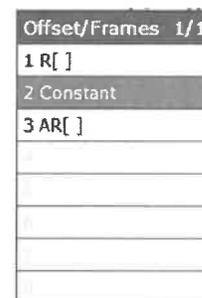
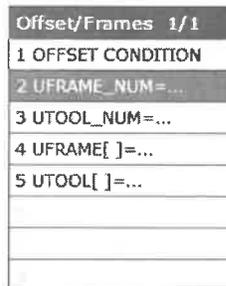
- 2 Press F1, INST (press NEXT if not displayed).



- 3 Select Offset/Frames on the second page of items.

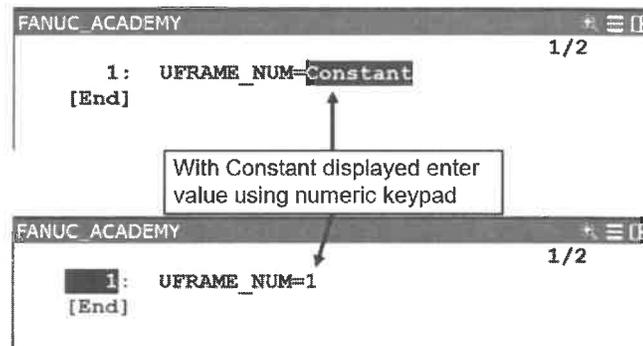


- 4 Select UFRAME_NUM=... or UTOOL_NUM=...
 - a Select the variable type of CONSTANT.



- 5 Enter the value of the frame being used.

11: Creating a Program



- 6 Repeat for the other frame setting as necessary.

NOTE: Adding these instructions will not set the desired frames, the lines either need to be executed or the frames manually set on the JOG MENU. Prior to recording motion instructions ensure the correct frames are set in the JOG MENU.

11.7.2 Default Motion Instructions

A list of default motion instructions is provided for your convenience. There are four default Motion Instructions predefined by the software, however these instructions can be modified according to your needs at any point. This section will specifically focus on selecting a default Motion Instruction.

NOTE: Modifying the items in the Default Motion Instructions will be covered in the next Chapter.

Anytime you record a position in a program you are doing so utilizing one of the four default Motion Instructions. The instruction that is populated in the program is dependent on the last default Motion instruction that was selected from the list.

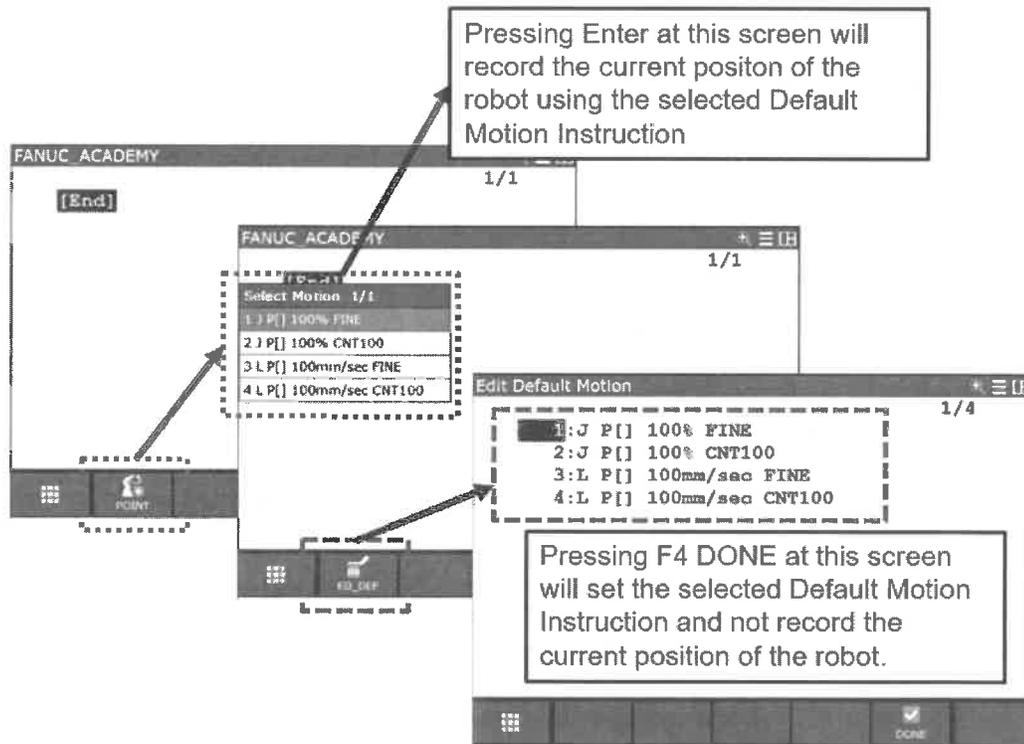
Procedure 11-4 Setting the Default Motion Instruction and Recording Positions

- 1 Press the SELECT key.
 - a Choose a program to edit from the list or create a new program.
- 2 Place your cursor on the line in the program you want to add your Motion Instruction.
- 3 Press F1 POINT.
- 4 If you **do not want** to record the current position of the robot, **skip** to the next step. If you want to record the current position of the robot continue with these steps.
 - a On the default Motion Instruction pop up menu place your cursor on the desired motion instruction and press ENTER. This will record the current position of your robot in the program **and** set the default Motion Instruction.
 - b Skip to Step 6.

NOTE: Once you press ENTER you will record the robots position, if there was pre-existing instructions on the line of the cursor they will be **overwritten**.

11: Creating a Program

- 5 To select a default Motion Instruction without recording the position of the robot.
 - a With the defaults pop up menu displayed press F1 ED_DEF.
 - b Place your cursor next to the desired default Motion Instruction.
 - c Press F4 DONE.
- 6 The default Motion Instruction has been set. Until changed this is the instruction that all positions will be recorded with.



- 7 To record a position using the current default motion instruction:
 - a Jog the robot to the location in the workcell where you want to record the Motion Instruction.
 - b Press and hold in the SHIFT key and press F1, POINT. The instruction will be added to the program automatically.

11.8 Testing a Program

Prior to executing any type of program in automatic mode they must be tested. FANUC offers two ways to test the program; STEP Test and Continuous Test. The benefit of testing the program allows the operator to verify the taught positions and paths, identify any interference points, and test the program logic.

11: Creating a Program

While testing programs do not forget how the mode select switch impacts the motion:

T1

Program playback at 250mm/sec or less

T2

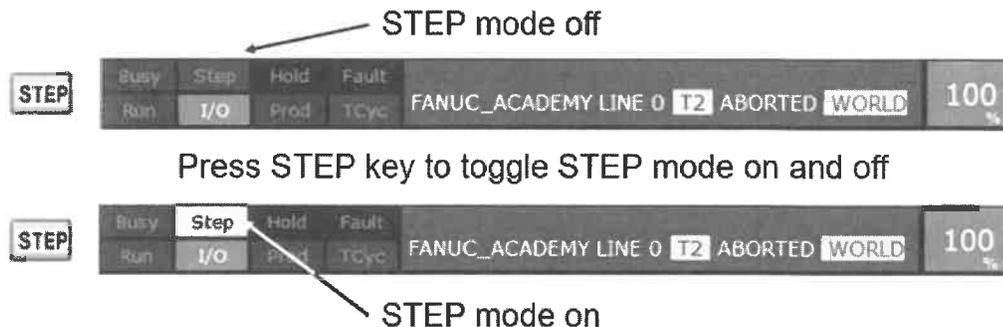
Program playback at programmed speed

11.8.1 Single STEP Testing a Program

Single STEP testing a program is when you test the program only one line at a time. This allows the operator to proof the motion and the logic that makes up the program with more control. Each line is executed one at a time and does not continue execution until prompted from the teach pendant.

In order to enter STEP Testing mode you must locate and press the STEP key on the teach pendant. After pressing the STEP key you will see the STEP status indicator change to yellow.

While testing you can test forward or you can also test the program backward. To move forward through the program you press the FWD key, to execute backward in the program press the BWD key.



Procedure 11-5 Executing a STEP Test

- 1 Enable the Teach Pendant
- 2 Select T1/T2 Mode
- 3 Choose the program to be tested
- 4 Place the cursor at the point to start testing
- 5 Press the STEP key (ensure you are in STEP mode)
- 6 Center the DEADMAN Switch
- 7 Press and hold Shift and press FWD (will execute just selected line)

NOTE: Shift must be held for the duration of motion, FWD should be pressed and released to execute the line.

- 8 When ready press FWD again to execute the next line

NOTE: At any time during testing you can release the shift key to perform a controlled stop,

11: Creating a Program

or release the DEADMAN switch to perform an EMERGENCY STOP.

When the teach pendant FWD and BWD keys are disabled and the teach pendant is enabled, "FBD" (Forward/Backward/Disabled) 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.

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.

11.8.2 Continuous Testing

A continuous test gives the operator the ability to test the program without stopping. This can give the operator a better understand and more accurate representation of each motion path.

You may see the programmed paths execute differently from the STEP test to the Continuous test. This is because of how the controller interprets the termination type with each type of test. In a STEP test every termination is treated like a Fine termination. This means the robot moves to the programmed position and pauses. However, in a continuous test each motion statement is executed based on how it is programmed. We will discuss termination types in the next chapter.

Procedure 11-6 Executing a Continuous Test

- 1 Enable the Teach Pendant
- 2 Select T1/T2 Mode
- 3 Choose the program to be tested
- 4 Place the cursor at the point to start testing
- 5 Check the STEP status indicator, ensure the controller is **not** in STEP mode
- 6 Center the DEADMAN Switch
- 7 Press and hold Shift and press FWD

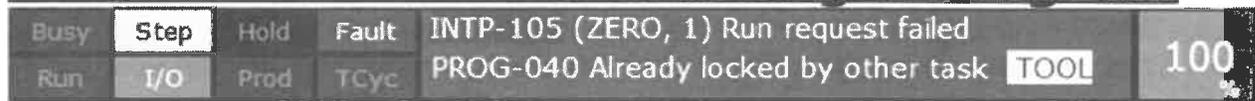
 **NOTE:** Shift must be held for the duration of motion, FWD should be pressed and released to execute the program

 **NOTE:** At any time during testing you can release the Shift key to perform a controlled stop, or release the DEADMAN switch to perform an EMERGENCY STOP.

11.8.3 Aborting a Program

If a previous program was tested and the [END] statement of the program was not executed you may need to abort this program in order to execute a different program. You will see a fault similar to below.

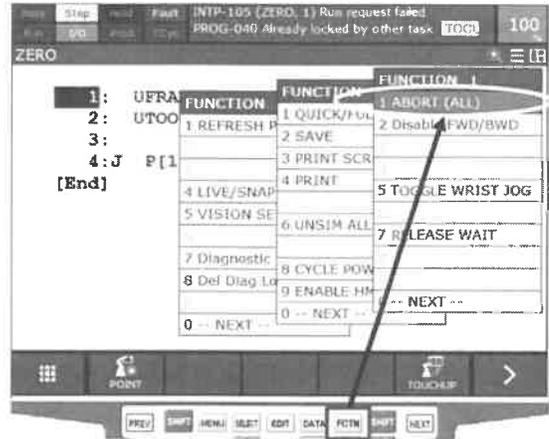
11: Creating a Program



All aborting a program does is end the previously selected program and set the newly selected as the current active program. You will not erase any program data or delete any programs when aborting.

Procedure 11-7 Aborting a program

- 1 Press FCTN
- 2 Select -1- ABORT (ALL).



11.8.4 Running a Program in AUTO

After conducting the STEP Test and Continuous test on the program from the Teach Pendant, running a program using the CYCLE START button allows the robot to run solo without any assistance from the user.

WARNING

Prior to executing a program in AUTO, ensure a successful STEP and Continuous test have been executed. For the first AUTO execution you should start the program at a lower general override speed and slowly approach the final general override speed. Failure to safely execute the program in AUTO Mode may result in injury to personnel or damage to equipment.

Procedure 11-8 Running a Program in AUTO

- 1 Disable the Teach Pendant
- 2 Select AUTO Mode
- 3 Choose the program to be tested
- 4 Place the cursor at the point to start the program
- 5 Check the STEP status indicator, ensure the controller is not in STEP mode
- 6 Press Reset to clear any faults
- 7 Press Cycle Start

11: Creating a Program

NOTE: When changing from T1/T2 to AUTO Mode there will always be a fault. Either; Teach Pendant Enabled in AUTO or Teach Pendant Disabled in T1/T2. This makes it necessary to press reset.

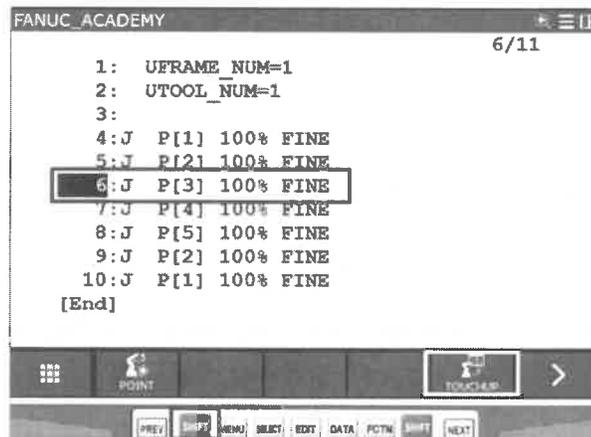
11.9 Touching Up a Point

While testing programs or running production it may be necessary to change positional data. This is referred to as touching up a point. When you touch up a point only the positional data is changed; the motion type, speed, and termination remains unchanged.

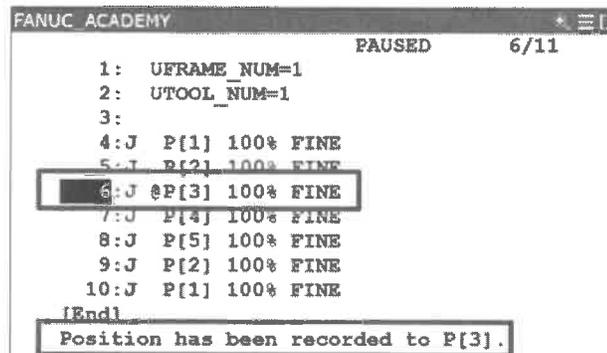
By changing only the positional data the path should remain largely unchanged. However, this does not preclude you from testing the change to ensure accuracy. Failure to properly test the change in the program may result in a crash or other undesirable outcomes.

Procedure 11-9 Touching Up a Point

- 1 Select the program that requires the edit.
- 2 Place the cursor on the line that needs to be touched up.
- 3 Manually jog the robot to the new position.
- 4 Press and hold Shift and press F5 TOUCH UP.



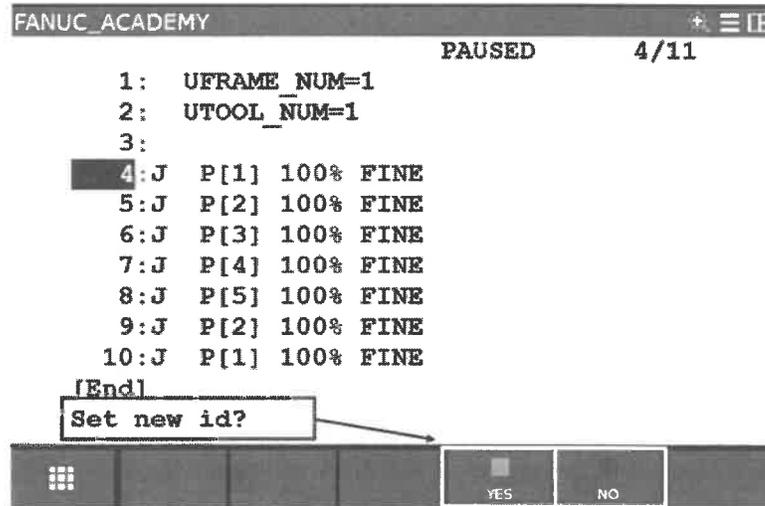
- 5 A successful touch up will display the @ symbol next to the position.



11: Creating a Program

11.9.1 Set New ID

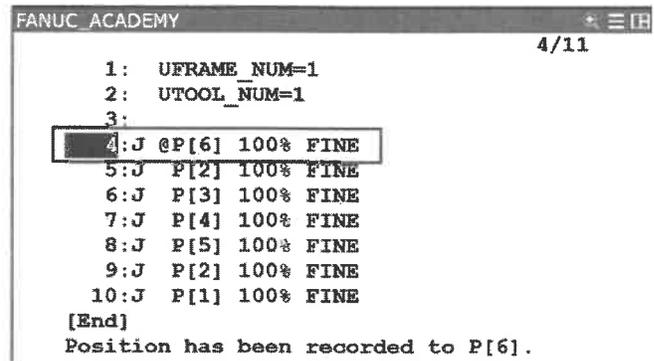
If you see the message Set New ID along the bottom of the teach pendant when touching up a point this is the software alerting you that multiple instances of this Position ID exist



The options are:

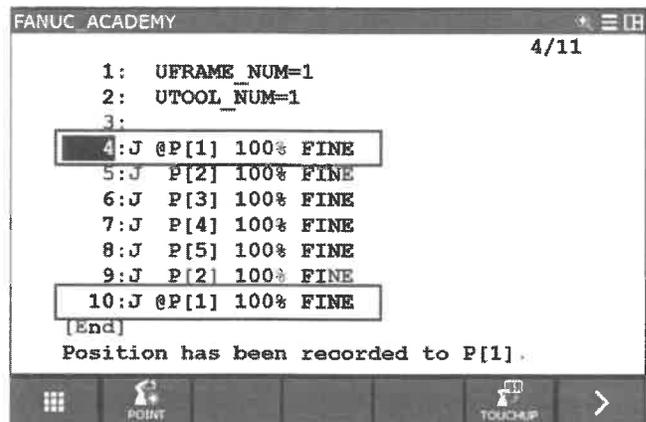
F[4] YES:

Position ID of the current line will change to the next available ID and the point is touched up.



F[5] NO:

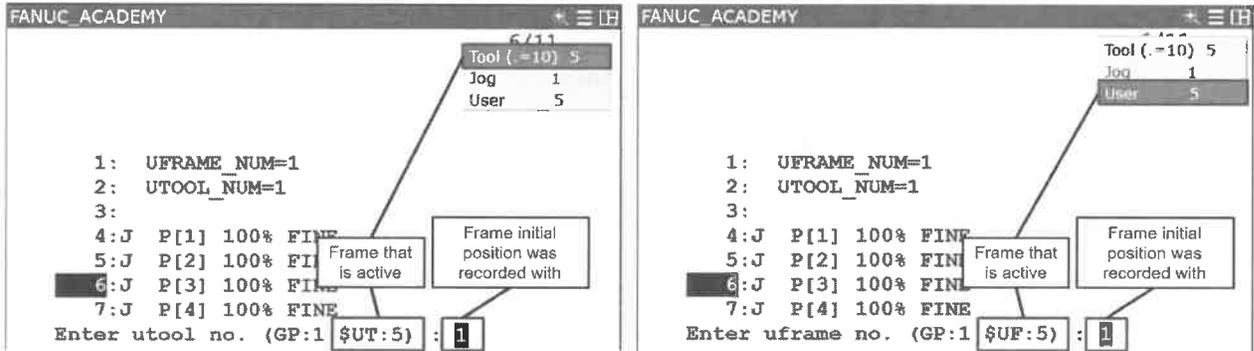
Position ID of the current line will not change, all position data with the same ID within the program will be touched up.



11: Creating a Program

11.9.2 Enter UTool / UFrame

If the message “Enter UTool or Enter UFrame” appears along the bottom of the teach pendant when touching up a point, this is the software alerting you of a frame inconsistency. When touching up a point, the USER Frame and TOOL Frame numbers in the Jog Menu must match the frames numbers used inside the program.



Procedure 11-10 Touching Up with the Enter UTool No. / Enter UFrame No. Prompt

- 1 Verify the correct frame to set
- 2 Type the value on the prompt line at the bottom of the screen.
- 3 Press Enter.

11.10 Saving and Restoring Program Files

Once a program is created, a file with the program name and a .TP extension will be saved to internal memory. For example, the program we have been using in this chapter was named FANUC_ACADEMY, the file name for this program would be FANUC_ACADEMY.TP.

11.10.1 Saving Program Files

As previously covered in the File Manipulation Chapter, a complete File Backup will save all of SRAM to include each individual program. However, it is also possible to save just the individual program itself. When you save the program they can be saved as a .TP file and/or as a .LS file.

There are two ways to save a program as a .TP file or a .LS file; using the function menu or the Select screen.

Procedure 11-11 Saving Individual Programs as a .TP file using the Function Menu:

- 1 Set the Default Device, see Chapter 7.4.1 Setting the Default Device.
- 2 Format the card (if necessary), see Chapter 7.4.2 Formatting the Device.

Procedure 11-12 Saving Individual Programs as a .TP file using the Select Screen:

- 1 Set the Default Device, see Chapter 7.4.1 Setting the Default Device.
- 2 Format the card (if necessary), see Chapter 7.4.2 Formatting the Device.

11: Creating a Program

- 3 Create a Directory (if necessary), see Chapter 7.4.3 Subdirectories.
- 4 From the Select screen highlight the program to save and press ENTER.
- 5 With the program displayed on the EDIT screen.
 - a Press FCTN.
 - b -0- Next.
 - c On the second page highlight SAVE.
 - d Press Enter.

- 3 Create a Directory (if necessary), see Chapter 7.4.3 Subdirectories.
- 4 From the Select screen highlight the program to save, **do not press ENTER.**
- 5 With the program highlighted on the Select screen.
 - a Press Next.
 - b Press F4, SAVE AS
 - c Verify the path to save of the screen
 - d If the path is correct press F1, DO_SAVE.

Procedure 11-13 Saving Individual Programs as a .LS File using the Function Menu:

- 1 Set the Default Device, see Chapter 7.4.1 Setting the Default Device.
- 2 Format the card (if necessary), see Chapter 7.4.2 Formatting the Device.
- 4 Create a Directory (if necessary), see Chapter 7.4.3 Subdirectories.
- 4 From the Select screen highlight the program to save and press ENTER.
- 6 With the program displayed on the EDIT screen.
 - a Press FCTN.
 - b -0- Next.
 - c On the second page highlight PRINT.
 - d Press Enter.

Procedure 11-14 Saving Individual Programs as a .LS File using the Select Screen:

- 1 Set the Default Device, see Chapter 7.4.1 Setting the Default Device.
- 3 Format the card (if necessary), see Chapter 7.4.2 Formatting the Device.
- 3 Create a Directory (if necessary), see Chapter 7.4.3 Subdirectories.
- 5 From the Select screen highlight the program to save, **do not press ENTER.**
- 7 With the program highlighted on the Select screen.
 - a Press Next.
 - b Press F5, PRINT
 - c The program name will populate the name field, change if necessary (not recommended).
 - d With the appropriate name entered, press ENTER..

11.10.2 Restoring Program Files

There are two ways to Restore a .TP File; using the File menu or the Select screen.

Procedure 11-15 Restoring a .TP File from the File Menu:

- 1 Set the Default Device, see Chapter 7.4.1 Setting the Default Device.
- 2 Navigate to the Directory that contains the file (if necessary), see Chapter 7.4.3 Subdirectories.
- 3 Sort the Directory by .TP Files
- 4 Place your cursor on the program to restore.
 - a Press F3, LOAD.
 - b Verify to load by pressing F4, YES.
 - c You will see a prompt confirming the file load.

Procedure 11-16 Restoring a .TP File from the Select Screen:

- 1 Set the Default Device, see Chapter 7.4.1 Setting the Default Device.
- 2 Navigate to the Directory that contains the file (if necessary), see Chapter 7.4.3 Subdirectories.
- 3 From the Select screen press NEXT
 - a Press F3, LOAD.
- 4 Type in the name of the program to be loaded.
 - a It is not necessary to have the correct case, however the spelling must be correct.
 - b With the name typed in, press ENTER.
 - c The program will be loaded.

11.11 Chapter Review

1. **To display a program the user must first navigate to which screen?**
 - A Edit Screen
 - B Select Screen
 - C Setup Screen
 - D Program Screen
2. **Which type of instruction defines for the robot how to move, where to move, how fast to move, and how to stop?**
 - A Move Instruction
 - B Logic Instruction
 - C Motion Instruction
 - D Kinematics Instruction
3. **How many default motion instructions can be defined using Handling Tool software?**
 - A 1
 - B 2
 - C 3
 - D 4
4. **What type of test should be performed prior to running the program in continuous?**
 - A Auto Test
 - B Step Test
 - C Constant Test
 - D Running in continuous is fine

Lab 15

Creating a Simple Program

Student Name: _____

Assignment: The student will:

- Create a Handling Tool Program
 - Verify Program Header Information
 - Reference User Frames and Tool Frames
 - Record Positions in a Program
 - Step Test, Continuous Test, and Run a Program in Automatic Mode
-

Condition: Given a FANUC Robot and a controller

- Step:**
- 1 Create a program called A_BOX. (See Section 11.5 Creating a Program, pg. 163)
 - 2 In the program detail, add your instructor's first name in the comment field. (See Section 11.6 Program Detail Information, pg. 165)
 - 3 Reference **your** User Frame and Tool Frame at the beginning of the program. (See Section 11.7.1 Motion Instructions and Frames, pg. 171)

 **NOTE:** Ensure the correct User and Tool Frames are set in the Jog Menu as well. Your Frame values may be different than the example. Use the frames that were created from the previous chapters. **Below is an example of the code.**

A_BOX:

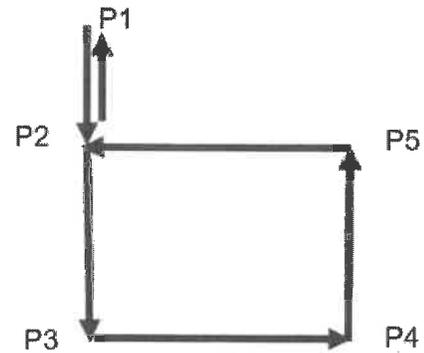
1: UFRAME_NUM=1

2: UTOOL_NUM=1

- 4 From the default motion instructions, set the "J P[] 100% FINE" instruction as the active default. (See Section 11.7.2 Default Motion Instructions, pg. 173)
- 5 Create a simple path that follows the shape as defined by your instructor.
 - a Ensure the program begins at a start position and returns to the initial position upon completion. The start position can be any position that is above the workpiece. We will change this in the next chapter.
 - b An example of the program is on the following page.

11: Creating a Program

```
1: UFRAME_NUM=1
2: UTOOL_NUM=1
3:J P[1] 100% FINE
4:J P[2] 100% FINE
5:J P[3] 100% FINE
6:J P[4] 100% FINE
7:J P[5] 100% FINE
8:J P[2] 100% FINE
9:J P[1] 100% FINE
```



NOTE: The above path is to describe the direction the program should flow, the robot motion will arc due to the Joint motion type used. Different motion types will be discussed in the following chapter.

- 6 With the program created perform a STEP Test. (See Section 11.8.1 Single STEP Testing a Program, pg. 175)
- 7 After a successful STEP Test perform a Continuous Test. (See Section 11.8.2 Continuous Testing, pg. 176)
- 8 After a successful Continuous Test run the program in Automatic Mode. (See Section 11.8.4 Running a Program in AUTO, pg. 177)

Completed:

Instructor: _____

11: Creating a Program

HandlingTool

Lab 16 Touching Up Points

Student Name: _____

Assignment: The student will:

- Touch Up a Point with Multiple Position ID's
- Touch Up a point and Set the Correct User Frame and Tool Frame

Condition: Given a FANUC Robot and a controller

-
- Step:**
- 1 Display the program A_BOX that was created in the previous lab.
 - 2 Touch Up **both** P [1]'s to be in a different location above the box. (**See Section 11.9 Touching Up a Point, pg. 178**)
 - 3 Bring up the Jog Menu and change both the TOOL and USER Frame values to a different number.
 - 4 Touch Up **only** the first P [2] to be located approximately two inches above its current location.
 - 5 With the program modified perform a STEP Test.
 - a If the program ran without faults continue to the next step.
 - b If there was a fault what was it? _____
 - c Correct the issue and test again.
 - 6 After a successful STEP Test perform a Continuous Test.
 - 7 After a successful Continuous Test run the program in Automatic Mode.

Completed:

Instructor: _____

11: Creating a Program

Lab 17

Manipulate Program Files

Student Name: _____

Assignment: The student will:

- Save the Program as a .TP File.
 - Save the Program as a .LS File.
 - Delete the Program File.
 - Restore the Program File.
-

Condition: Given a FANUC Robot and a controller

- Step:**
- 1 Set the default device.
 - a Set the path to the Directory named "INDIVIDUAL FILES".
 - 2 Save A_BOX as a .TP File. (**See Section 11.10.1 Saving Program Files, pg. 180**)
 - 3 Save A_BOX as a .LS File.
 - 4 From the Select Screen highlight A_BOX
 - a Press F3, Delete
 - b Verify YES to delete by pressing F4, YES.
 - 5 Restore A_BOX. (**See Section 11.10.2 Restoring Program Files, pg. 182**)

Completed:

Instructor: _____

12 MOTION INSTRUCTIONS

12.1 Chapter Objectives

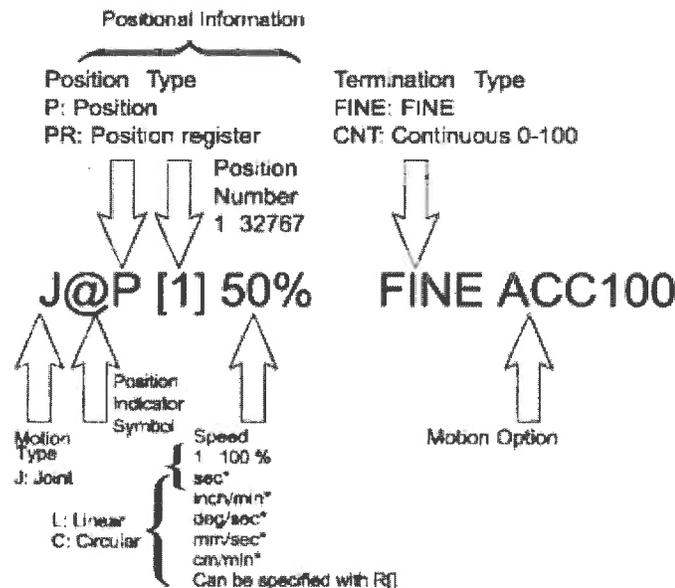
- Elements that describe a motion instruction
- Understand differences between Linear and Joint Motion
- Position Information Overview
- Understand the difference between a Position and a Position Register
- Understand Robot Speed Values
- Learn the difference between FINE and Continuous Termination

12.2 Motion Instruction Overview

Instructions that tell the robot where to go, how to move there, at what speed to move, and what to do when you get there are called motion instructions.

A motion instruction includes:

- Motion Type – How the robot moves to the position.
- Position indicator symbol – Indicates that the robot is at the taught position
- Positional Data – Where the robot moves.
- Speed – How fast the robot moves to a position.
- Termination Type – How the robot ends the move to the position.



NOTE: Motion Options will be covered in a later Chapter.

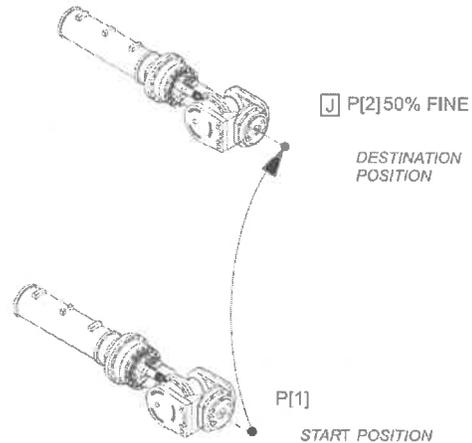
12: Motion Instructions

12.3 Motion Types

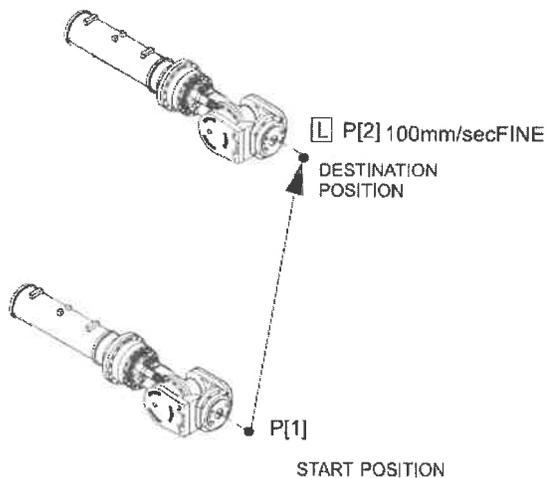
The Motion Type defines how the robot will move to the destination position. It is specified in the program not by how the robot was jogged to the point, but the motion type specified on the motion instruction. Depending on the controller and software there may be as many as four motion types: Joint, Linear, and Circular.

12.3.1 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
- Speed is specified as a percentage of the total default speed, or timed; milliseconds or seconds. The actual speed of the move is dependent on the speed of the slowest axis



12.3.2 Linear Motion

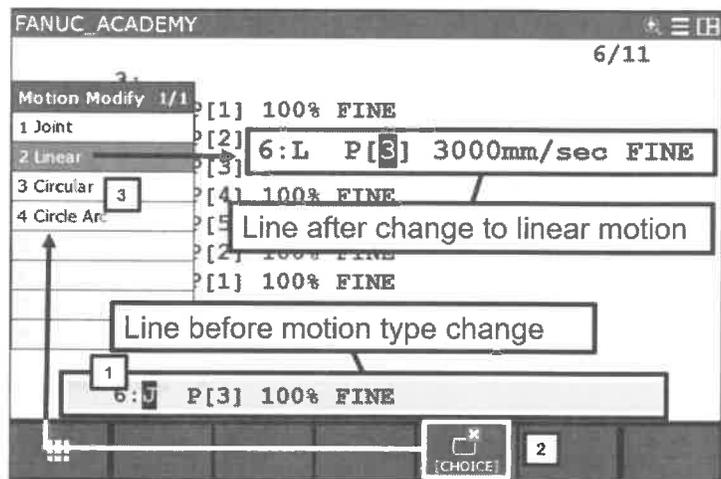


- Causes the robot to move the TCP in a straight line from the start position to the destination position.
- Is programmed at the destination.
- Speed is specified in velocity; millimeters per second, centimeters per second, inches per minute, degrees per second. Or can be timed; milliseconds, seconds.

12.3.3 Changing Motion Types

Procedure 12-1 Changing Motion Type

- 1 Highlight the motion type to change.
- 2 Press the F4, [CHOICE] key.
- 3 Choose desired motion type from the popup menu.



12: Motion Instructions

12.3.4 Circular Motion

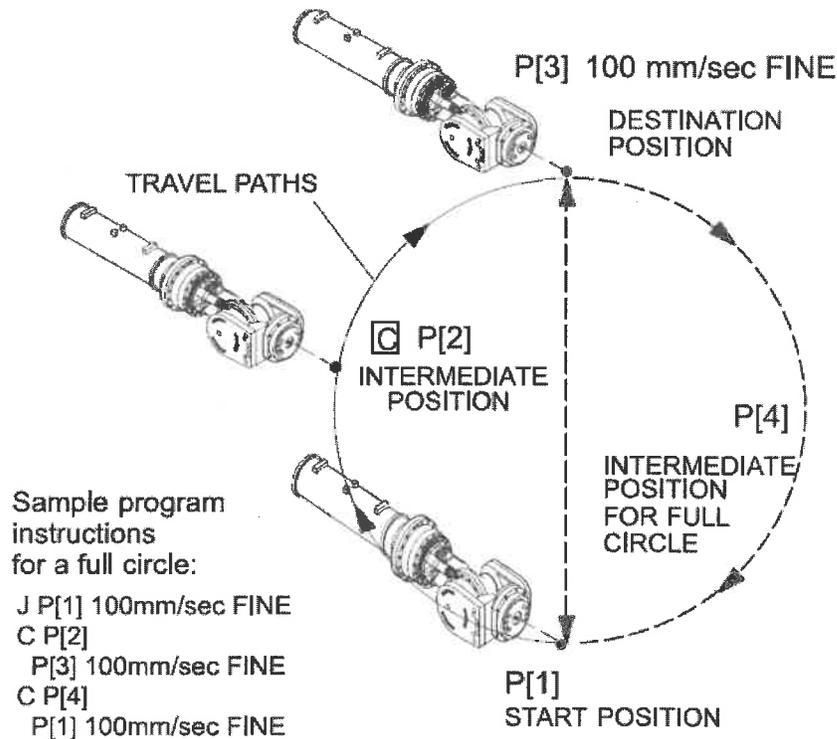
- Causes the robot to move the tool center point in an arc from the start position through an intermediate to the destination position.
- Is programmed at the intermediate position.
- Speed is specified in inches per minute, millimeters per second, and centimeters per minute.

When you add a motion instruction that has circular motion type, the following appears on the screen:

```
C P[2]
P[3] 100mm/sec FINE
```

The first position, P [2] in the example above, is the intermediate position. The intermediate position is automatically recorded 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, F3.

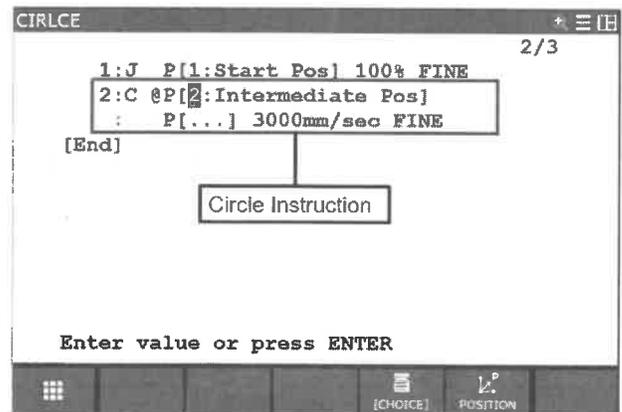
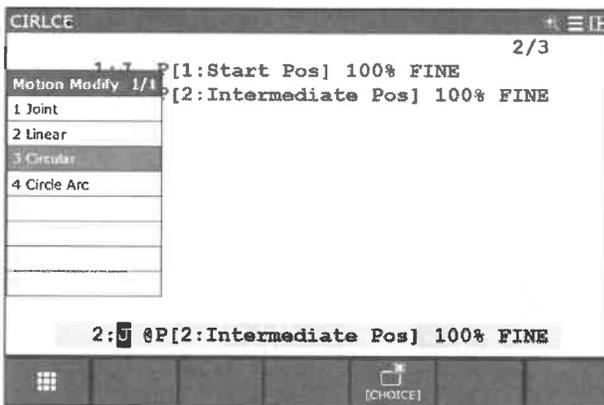
When creating a circle the programmer must teach two half circles. Each half circle consists of an intermediate position and destination position. Below is an example of what a programmed complete circle should look like.



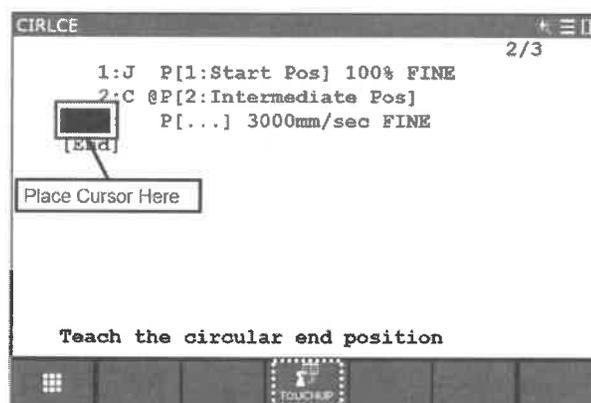
12: Motion Instructions

Procedure 12-2 Programming a Circle

- 1 Move the robot to starting point of the circle.
 - a Record this position as a Joint or Linear move.
 - b This first position will move the robot from wherever it is in the cell to the starting point of the circle.
- 2 Move the robot to the first intermediate position of the circle.
 - a Record this position using the same default motion instruction as the first position.
 - b Highlight the motion type of the intermediate position.
 - c Press F4, [CHOICE].
 - d Select Circular.
 - e The instruction will change to the circular instruction.



- 3 Move the robot to the destination position.
 - a Place your cursor as shown below.
 - b Touch up this to record the destination position.

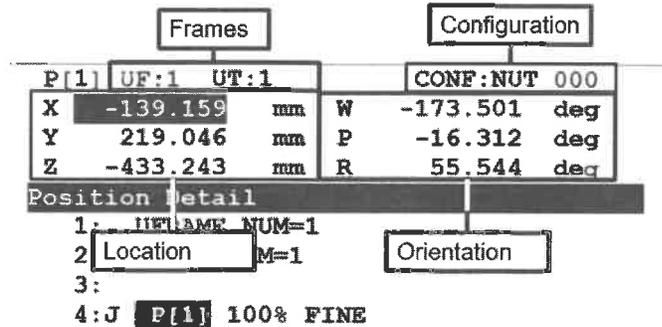


- 4 Repeat **Step 2** and **Step 3** for the second half of the circle if necessary.

12: Motion Instructions

12.4 Position Information

Positional information describes the location, orientation, and configuration of the tool center point when a motion instruction is executed within a program. All of the positional information is recorded when the motion instruction is added to the program.



Item	Description
UF (User Frame)	References the User Frame the position was recorded with respect to
UT (Tool Frame)	References the Tool Frame the position was recorded with respect to
Configuration	References the posture of the robot upon arrival at the position.
Location	Describes the three-dimensional location of the position.
Orientation	Describes rotation about X (Yaw), rotation about Y (Pitch), and rotation about Z (Roll).

12.4.1 Configuration

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.

	J1 J4 J6
(F L U T)	(0 , 0 , 0)
Joint Placement	Turn Number

Joint Placement

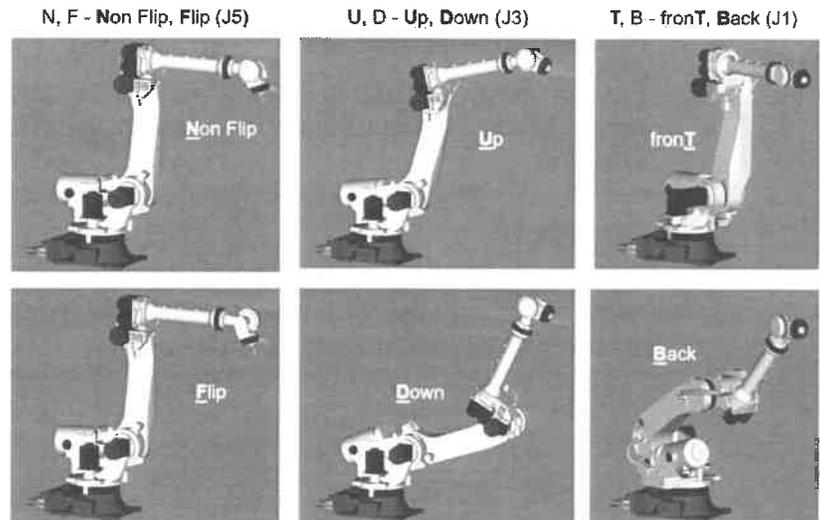
- { FLIP Upward and downward placement of the wrist
- { NOFLIP
- { LEFT Right and left placement of the arm
- { RIGHT For horizontally articulated robots only
- { UP Upward and downward placement of the arm
- { DOWN
- { FRONT Backward and forward placement of the arm
- { BACK

Turn number

- { 1: 180° ~ 539°
- { 0: -179° ~ 179°
- { -1: -539° ~ -180°

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 full articulated robots (4 vs 6 axes robots)

- **Wrist:** N, F - Non Flip, Flip
- **Elbow:** U, D - Up, Down
- **Robot:** T, B - Front, Back



12.5 Position Types

There are two types of positions that store the position information.

- **Position:** Local
- **Position Register:** Globally Accessible

In the 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.

Both types of positions contain the same information as annotated in **Section 12.4 Position Information**. The difference between the two is how they are used and where the information is stored within the software.

12.6 Position

When a program point is recorded, by default, it is recorded as a position and identified with a “P” and an associated ID Number, it will show in the program as P [1]. A position is referred to as Local, meaning it is defined locally within the current program.

It is possible to have multiple positions identified within the same program by the same ID number. In this case all positions within the same program contain the exact same positional data.

It is also possible to have the same position ID across multiple programs. In this case the positional data will be different. This is because a position only exists within the program that it is recorded in (Local). There is no direct correlation between P [1] in one program and P [1] in a different program.

12.7 Position Registers

Position Registers can be used to store globally accessible positions. These positions are predefined on the Position Register Data Screen and can be shared for use in many programs. Generally a Position Register is used for positions that will be used multiple times within a routine such as Home, Pick Positions, Drop positions, etc.

A Position Register is identified within the program with a “PR” designation and an ID Number, it will appear as PR [1] for example.

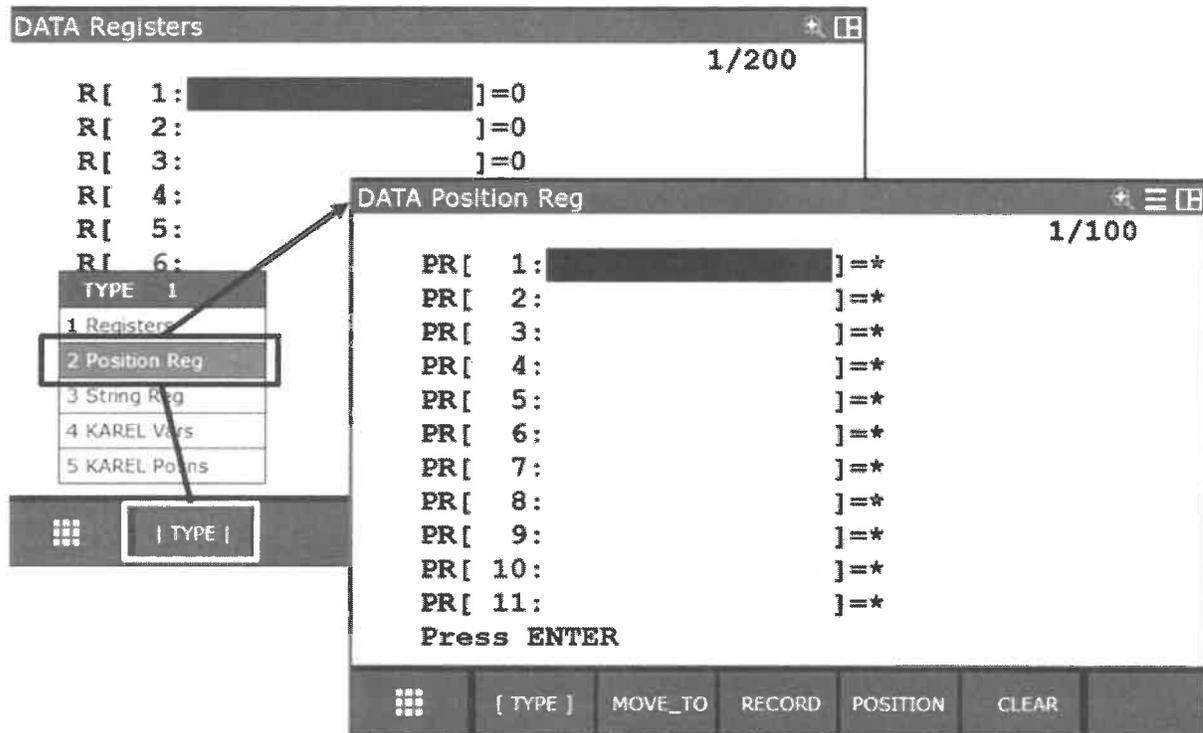
Position Register Data is globally accessible, this means that the **positional information*** will be same anywhere that the same Position Register ID Number is seen within any program.

CAUTION

Position Register data is shared between all programs that contain the specified Position Register. Although the positional information will be the same, depending on the frames referenced the physical location may vary.

Procedure 12-3 Accessing Positon Register Data

- 1 Press DATA
- 2 Press F 1,[TYPE].
- 3 Choose Position Registers



12.7.1 Example of Position Registers

- HOME Position:** The home position 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.
- 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.
- Safe Position:** The safe position 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.
- 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.

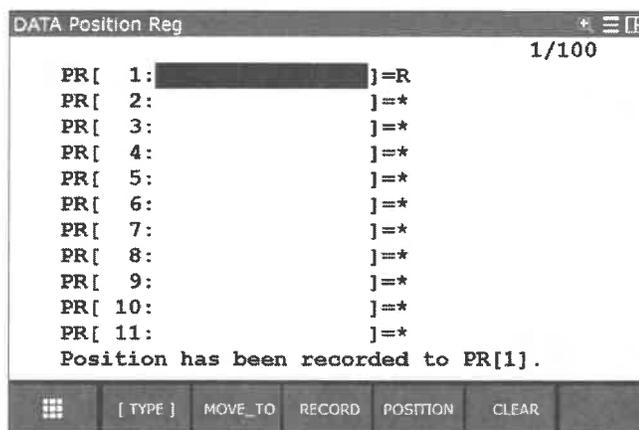
12: Motion Instructions

12.7.2 Creating Position Registers

Position Registers are stored and recorded from the Position Register Data Screen.

Procedure 12-4 Creating a New Position Register

- 1 Move the Robot to the position to record
- 2 Highlight the PR to record
- 3 Name the Position Register
- 4 Press Shift and F3, RECORD
- 5 You will see an "R" on the PR line

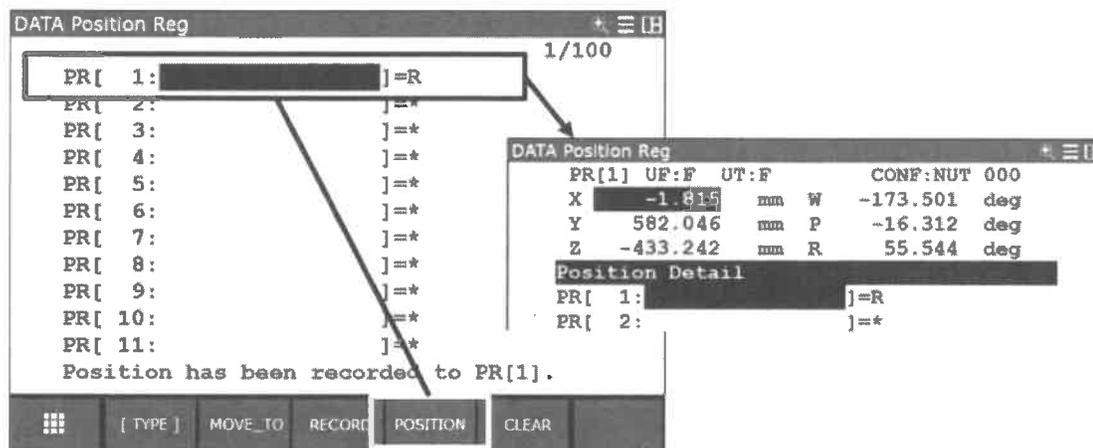


NOTE: Ensure the correct frames are set prior to recording Position Registers.

12.7.3 Viewing Position Register Data

Procedure 12-5 Viewing Position Register Data

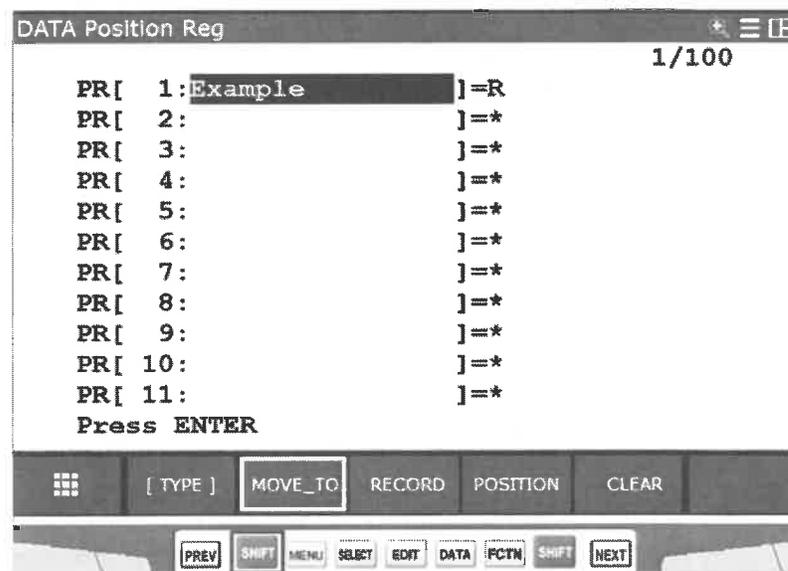
- 1 Highlight the PR to be viewed
- 2 Press F4, POSITION



12.7.4 MOVE TO a Position Register

Procedure 12-6 Using MOVE TO a Position Register

- 1 Highlight the PR to move to.
- 2 Set the Mode to T1
- 3 Enable the Teach Pendant
- 4 Center the DEADMAN Switch
- 5 Clear any faults
- 6 Press and Hold Shift
- 7 Press F[2] MOVE TO



CAUTION

Before executing a MOVE TO, and during the move, ensure the robot does not collide with any obstacles. The robot WILL take the most direct path to the recorded position.

12.7.5 Adding a Position Register to a Program

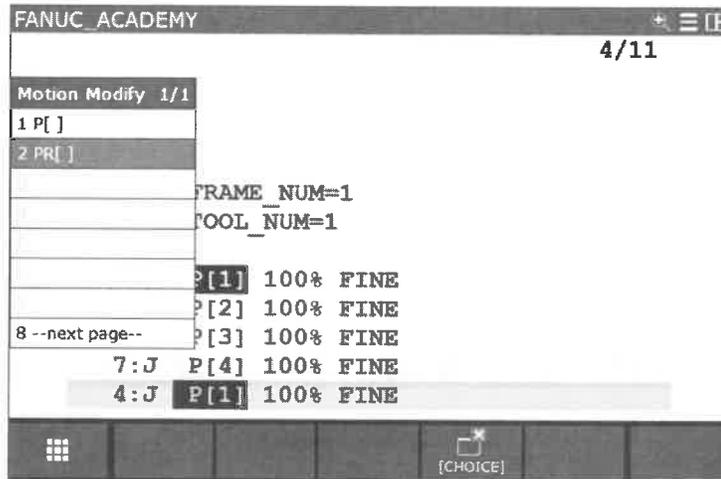
When recording program points, the default recording is a position. If you want to add Position Registers to an existing program or a new program they must be manually added by altering the existing motion statement.

Procedure 12-7 Adding a Position Register to a Program

- 1 Record a bogus motion statement.
 - a The physical location of the position is arbitrary as we will overwrite this data with the data stored in the Position Register.
- 2 Highlight the position field.

12: Motion Instructions

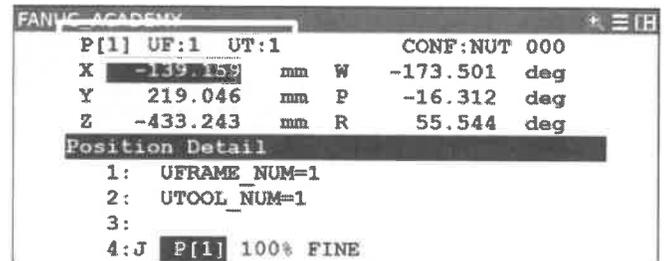
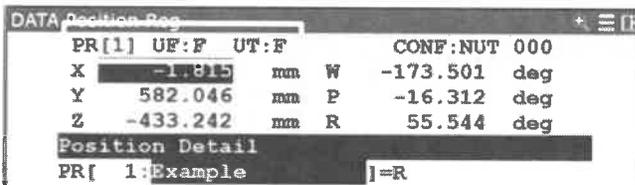
- 3 Press F 4, [CHOICE].
- 4 Choose PR [] from the list.



- 5 Manually enter the ID of the Position Register.

12.7.6 Position Register Representation

As previously discussed a Position Register is a globally accessible position. It is important to note the verbiage used to describe a position register, again globally accessible. This does not mean that the physical location of the Position Register for any given ID will always be in the same spot.



Notice, on the left in the image above, the frames associated with PR [1] are UF: F and UT: F. However, on the right, the frames associated with P [1] are listed as UF: 1 and UT: 1. How frames are used with Position Registers allow them to become more versatile and globally accessible. It also explains why a Position Register may not always be in the same physical location all the time.

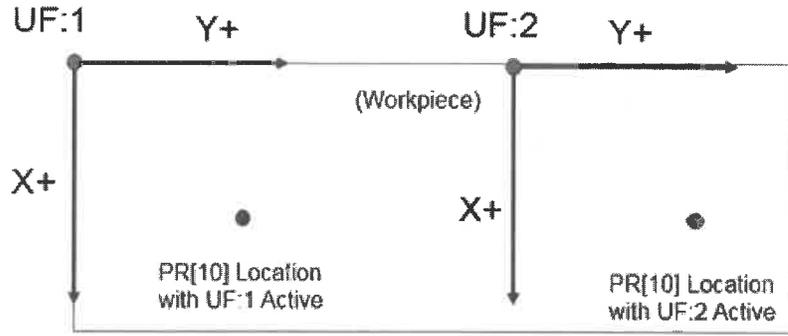
The USER frame, UF:F, and the TOOL frame, UT:F, will be set to 15 (FHex), which indicates that the currently active USER frame and TOOL frame will be used. What this means is that the controller will apply the position data of the Position Register to whatever frames are currently active and move to the stored position.

In essence Position Registers are not restricted by setting a specific frame in order to utilize them. This also means that if the incorrect frames are active utilizing a position register may result in a crash or undesirable outcomes.

12: Motion Instructions

The relationship between Position Registers and Frames is illustrated below. As previously stated, a Position register will use the active USER and TOOL Frame combination and apply the data to those references.

Below we can see two USER Frames with their respective origins annotated. The Position Register data is applied to the USER Frame origins and the robot position would be as seen. Notice, the same Position Register is used however the physical position differs.



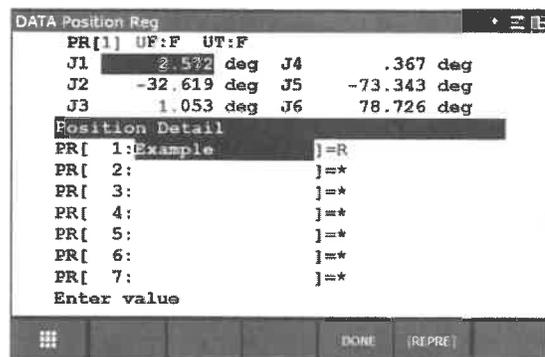
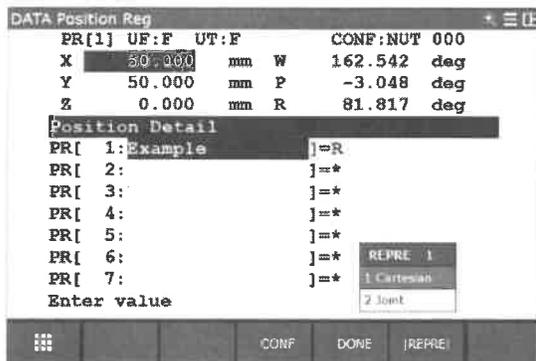
PR[10]	UF:F	UT:F	CONF:NUT	000
X	50.000	mm W	162.542	deg
Y	50.000	mm P	-3.048	deg
Z	0.000	mm R	81.817	deg

12.7.7 Creating a Truly Global Position Register

If there is a requirement for a Position Register to be in the same physical location all the time, regardless of active USER and/or TOOL Frame combination, this can be accomplished by changing the positional representation. If the positional data is represented in degrees this makes the position an absolute position. With this representation, regardless of the set USER and TOOL Frames the controller will send the Robot to the same location every time because the position is based on degrees at each joint.

Procedure 12-8 Changing Position Register representation

- 1 Access the Position Register positional data
- 2 Press F5, [REPRE]
- 3 Select Joint
- 4 Select Continue from the prompt box



12: Motion Instructions

12.8 Speed

Speed dictates how fast the robot will move while executing motion. Speed can be manipulated in two ways; as part of the motion statement or from the general override.

12.8.1 Motion Statement Speed Values

Speed must be specified as part of the motion statement. When assigning speed in the motion statement the available units are reflective of the motion type selected.

JOINT Motion Type

Motion Modify 1/1
1 %
2 sec
3 msec

LINEAR / CIRCULAR Motion Type

Motion Modify 1/1
1 mm/sec
2 cm/min
3 inch/min
4 deg/sec
5 sec
6 msec

Procedure 12-9 Changing Speed Values

- 1 Highlight the speed
- 2 Type in the new value
- 3 Press Enter

Procedure 12-10 Changing Speed Units

- 1 Highlight the speed
- 2 Press F[4] CHOICE
- 3 Choose the new unit from the list

12.8.2 General Override

The general override is located at the top right corner of the teach pendant in the green box. The general override identifies the percentage of speed available for motion. To manipulate the general override utilize the “+%” and “-%” keys located on the bottom of the teach pendant.

- Can change speeds while running with the “+%” and “-%” keys, changes are 5% increments by default.
- Speeds range from .01% (VFine: Incremental) to 100%.
- Is a percentage of speed allowed.

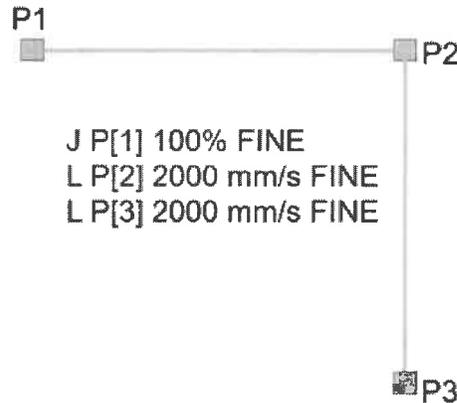
An example of how the General Override impacts speeds is if the motion statement is programmed to run at 100% and the general override is set to 75%, the robot will operate at 75% of the programmed 100%. For linear speeds, if the motion statement is programmed to execute at 1000 mm/sec and the override is set to 75%, the robot will operate a 750 mm/sec.

12.9 Termination Type

Termination type specifies the robot's action as it approaches the end of the motion and is programmed at the destination. Termination type can be specified as Fine and Continuous.

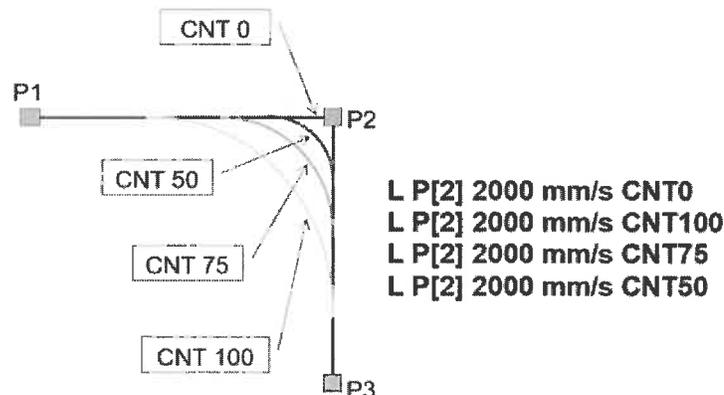
12.9.1 Fine Termination

Fine termination type causes the robot to stop at the destination position before moving to the next position. The robot's tool tip accelerates to the defined speed, and then decelerates as it approaches the recorded position and comes to a complete stop, before continuing to the next recorded position.



12.9.2 Continuous Termination

Continuous (CNT) termination type allows the robot to decelerate as it approaches the destination position but does not stop at the recorded position before it accelerates toward the next position. In the examples shown below, 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 while at CNT100 the robot is farthest, with minimum deceleration.

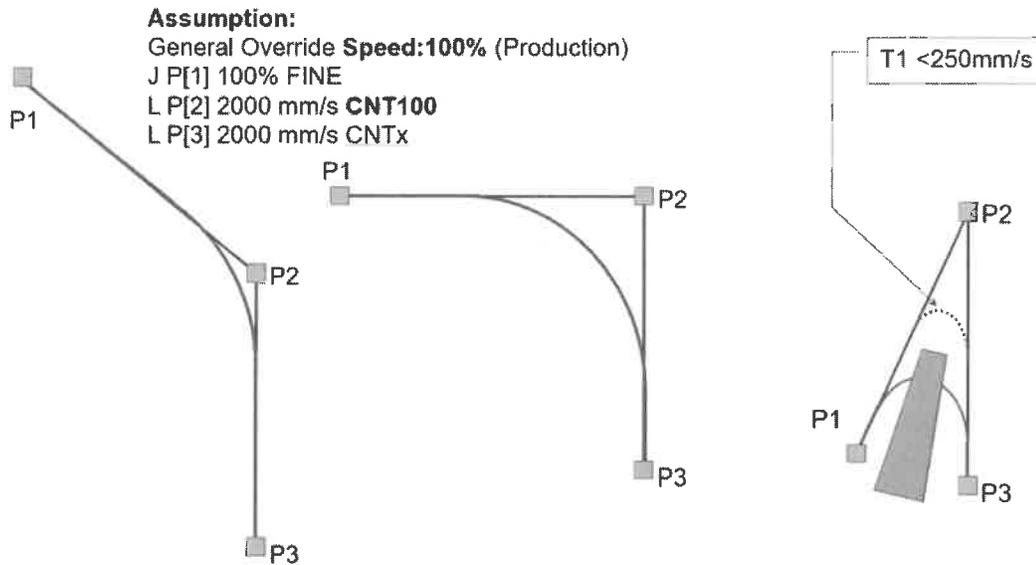


WARNING

There may be inconsistencies between the T1 test and running in Auto. Run the programs in Auto at slow speeds at first to verify the path.

12: Motion Instructions

The severity of the directional change at each taught position and the speed of the program determine how the Continuous rate will affect the path. As the angle increases the distance from the taught position increases as well.

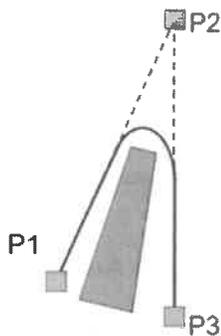


NOTE: The Constant Path software option (R663) allows the robot to follow the programmed path regardless of override rate. The far right above example shows the robot path in T1 compared to Auto Mode without Constant Path.

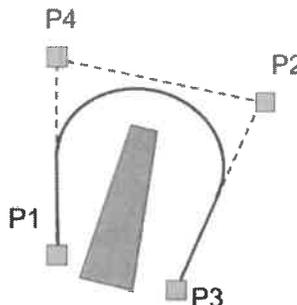
If corrections are required for the path there are three main approaches for correction; lower the termination rate or the speed on the position (left), add a position and touchup the old position (middle), or touchup the affected position farther away from the current point (right).

General Override **Speed:100%** (Production)

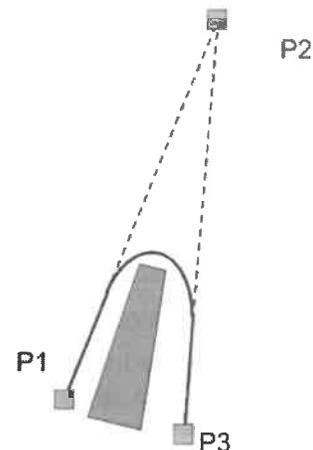
J P[1] 100% FINE
 L P[2] **2000** mm/s **CNT60**
OR
 L P[2] **1200** mm/s **CNT100**
 L P[3] 2000 mm/s FINE



J P[1] 100% FINE
ADD P[4]
 L **P[4]** 2000 mm/s **CNT100**
 L P[2] 2000 mm/s **CNT100**
 L P[3] 2000 mm/s FINE



J P[1] 100% FINE
 L P[2] 2000 mm/s **CNT100**
 L P[3] 2000 mm/s FINE



12.10 Chapter Review

1. Which motion type moves the TCP in a straight line from the start position to the destination position?

- A Linear
- B Joint
- C Circular
- D Spline

2. What is the name of the position in the square below?

```
c P[2]  
F[3] 100mm/sec FINE
```

- A Start Position
 - B Intermediate Position
 - C Destination Position
 - D First Position
3. Fine termination type causes the robot to _____ at the destination position before moving to the next position.
- A Slow down to 50%
 - B Accelerate
 - C Stop
 - D Bypass

12: Motion Instructions

HandlingTool

Lab 18

Creating Position Registers

Student Name: _____

Assignment: The student will:

- Create a Position Register for Home, Repair, and Safe.
-

Condition: Given a FANUC Robot and a controller

- Step:**
- 1 Create a Position Register for a Home Position (PR [1]). (See Section 12.7.2 Creating Position Registers, pg. 196)
 - 2 Label the new Position Register with an appropriate name.
 - 3 Convert the data to Joint Representation. (See Section 12.7.7 Creating a Truly Global Position Register, pg. 199)
 - 4 Move the robot away from the position.
 - a Using the MOVE TO key, move the robot back to the recorded Position Register. (See Section 12.7.4 MOVE TO a Position Register, pg. 197)
 - 5 Perform Steps 2-4 to create a Repair Position (PR [2]) and a Safe Position (PR [3]).

Completed:

Instructor: _____

12: Motion Instructions

Lab 19

Modify a Program

Student Name: _____

Assignment: The student will:

- Modify Motion Instructions
-

Condition: Given a FANUC Robot and a controller

- Step:**
- 1 Display the program A_BOX.
 - 2 Modify the Motion Instructions to change how the program operates.
 - a The only rule, you cannot use the same speed value or termination value more than once.

 **NOTE:** Below is an example, you can follow the example or make your own changes.

- ```
1: UFRAME_NUM=1
2: UTOOL_NUM=1
3:J P[1] 100% FINE
4:J P[2] 88% CNT1
5:L P[3] 250mm/sec CNT50
6:J P[4] 50% CNT100
7:L P[5] 1000mm/sec CNT40
8:L P[2] 1988mm/sec CNT0
9:J P[1] 100% CNT75
```
- 3 STEP test the new path
  - 4 Continuous Test the new Path
  - 5 Run the modified program in Automatic Mode

Completed:

Instructor: \_\_\_\_\_

## Lab 20

### Create a Shapes Program

---

Student Name: \_\_\_\_\_

**Assignment:** The student will:

- Create a Program that traces three distinct shapes.
  - Become familiar with Motion Instructions
- 

**Condition:** Given a FANUC Robot and a controller

---

- Step:**
- 1 Create a new program and name it "A\_SHAPES". (See Section 11.5 Creating a Program, pg. 163)
  - 2 Program the robot to trace the square, circle, and triangle from the template provided by your instructor.
    - a Ensure you reference the USER and TOOL Frame combination at the top of the program. (See Section 11.7.1 Motion Instructions and Frames, pg. 171)
    - b The robot MUST start and end each shape at the HOME Position Register.
  - 3 For the Square portion create a Default motion instruction as follows: (See Section 11.7.2 Default Motion Instructions, pg. 173)  

```
L P [] 750 mm/sec CNT50
```
  - 4 For the Circle portion create a Default motion instruction as follows: (See Section 11.7.2 Default Motion Instructions, pg. 173)  

```
L P [] 1000 mm/sec CNT100
```
  - 5 For the Triangle portion create a Default motion instruction as follows: (See Section 11.7.2 Default Motion Instructions, pg. 173)  

```
L P [] 500 mm/sec CNT 75
```

 **NOTE:** For cycle time considerations, the move from HOME to the shape and from the shape to HOME should be JOINT motion, 100% Speed, with a USER defined termination.

- 6 See the following page for complete program syntax.

# 12: Motion Instructions

```
1: UFRAME_NUM=1
2: UTOOL_NUM=1
3: J PR[1:HOME] 100% FINE
4: J P[1:Square] 100% CNT50
5: L P[2] 750mm/sec CNT50
6: L P[3] 750mm/sec CNT50
7: L P[4] 750mm/sec CNT50
8: L P[1:Square] 750mm/sec CNT50
9: J PR[1:HOME] 100% FINE
10: J P[5:Circle] 100% FINE
11: C P[6]
 :P[7] 1000mm/sec CNT100
12: C P[8]
 :P[5:Circle] 1000mm/sec CNT100
13: J PR[1:HOME] 100% FINE
14: L P[9:Triangle] 500mm/sec CNT75
15: L P[10] 500mm/sec CNT75
16: L P[11] 500mm/sec CNT75
17: L P[9: Triangle] 500mm/sec CNT75
18: J PR[1:HOME] 100% FINE
```

- 7 STEP Test, Continuous Test, then run the program in Automatic Mode.

Completed:

Instructor: \_\_\_\_\_

## Lab 21

### Manipulate Individual Files

---

Student Name: \_\_\_\_\_

**Assignment:** The student will:

- Save A\_SHAPES as a .TP File.
  - Save A\_SHAPES as a .LS File.
  - Save Position Register Data.
- 

**Condition:** Given a FANUC Robot and a controller

---

- Step:**
- 1 Set the default device.
    - a Set the path to the Directory named "INDIVIDUAL\_FILES".
  - 2 Save A\_SHAPES as a .TP File. (See Section 11.10.1 Saving Program Files, pg. 180)
  - 3 Save A\_SHAPES as a .LS File. (See Section 11.10.1 Saving Program Files, pg. 180)
  - 4 Navigate to the Position Register Screen
    - a Press FCTN.
    - b Press -0- Next.
    - c Highlight SAVE.
    - d Press ENTER.
  - 5 View the Directory from the file screen and verify POSREG.VR is present.

Completed:

Instructor: \_\_\_\_\_

**13 EDIT COMMAND****13.1 Chapter Objectives**

- Inserting blank lines into a program.
- Deleting lines from a program.
- Copying and Pasting Instructions within a program.
- Replacing items within a program.
- Renumbering Position ID's.
- Displaying Comments within a program.
- Undo commands.
- Remarking lines in a program.
- Displaying color backgrounds
- Displaying I/O status.

**13.2 Edit Command Menu**

The Edit Command Menu allows the programmer access to items to change motion and non-motion instructions. Edit Command is located on the second page of items from the EDIT screen under the F5, [EDCMD] key.

The items within Edit Command consist of the following:

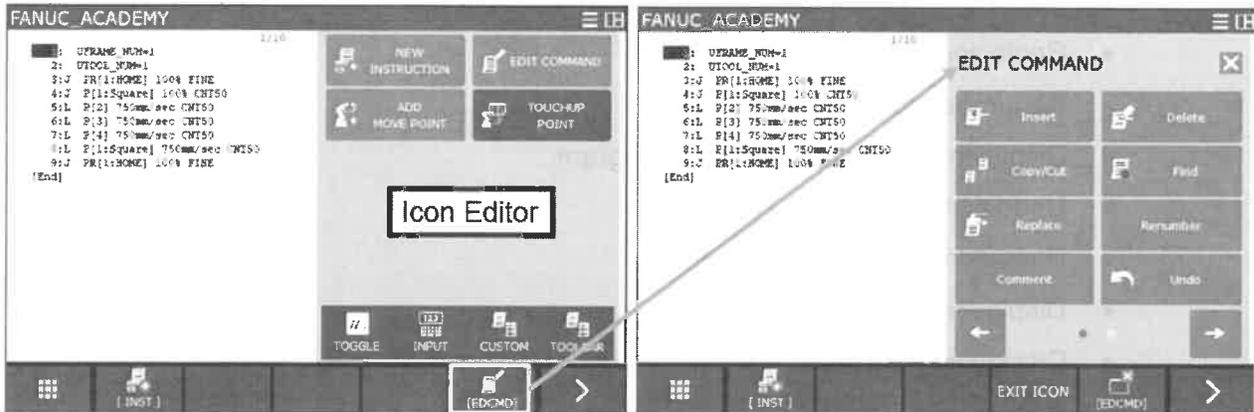
| Item       | Description                                                                                                                     |
|------------|---------------------------------------------------------------------------------------------------------------------------------|
| Insert     | Adds blank lines to the program. Must be used when adding positions or instructions to a program. Otherwise you will overwrite. |
| Delete     | Removes line(s) from a program                                                                                                  |
| Copy/Cut   | Can copy a range of lines and paste them in the same or another program                                                         |
| Find       | Allows searching for an instance of an instruction                                                                              |
| Replace    | Allows search and replace of instructions or variable items                                                                     |
| Renumber   | Renumbers positions (not PR's) sequentially from start to finish in a program                                                   |
| Comment    | Hides or displays comments from other tables on the EDIT Screen                                                                 |
| Undo       | Allows an undo of the latest edit only                                                                                          |
| Remark     | Masks out in instructions in a program                                                                                          |
| Color      | Highlights remarks with a yellow background and I/O Status with Red (OFF) or Green (ON).                                        |
| I/O Status | Displays or masks current status of I/O utilized in a program                                                                   |

# 13: Edit Command

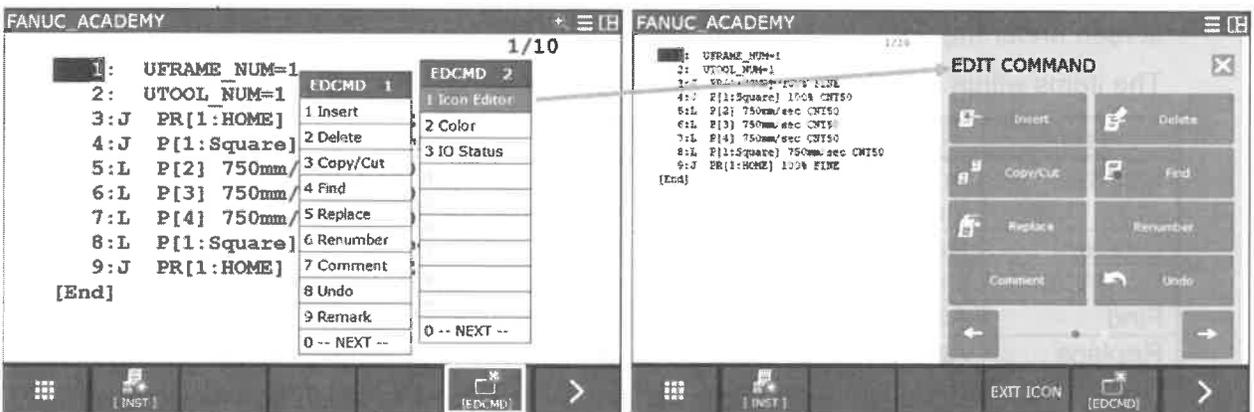
## 13.2.1 Accessing the Edit Command Menu

There are two ways to access the items on the Edit Command menu, either through the Icon Editor or the F5, [EDCMD] soft key. Regardless of the method to access the items on the Edit Command menu the options are the same.

- To access the Edit Command menu using the Icon Editor press F5, [EDCMD]. You should see a screen similar to below.



- If the Icon Editor is **not displayed**, press F5, [EDCMD] and choose Icon Editor from the menu. You should see a screen similar to below.



- If you prefer to use the Edit Command fly out menu simply press F4, EXIT ICON and navigate the menu as shown in the above image at the left.

**NOTE:** Both R30iB PLUS and R30iB software feature the Icon Editor, the only difference is how they are displayed. The above procedures will work for either software.

## 13.3 Inserting Blank Lines

Inserting blank lines into a program accomplishes two objectives; provides some eye relief to the programmer/operator by breaking up line(s) of code and provides space to add code.

On FANUC robots, when you add instructions to a program it does not automatically insert the new instruction, it will overwrite whatever was on the existing line.

# 13: Edit Command

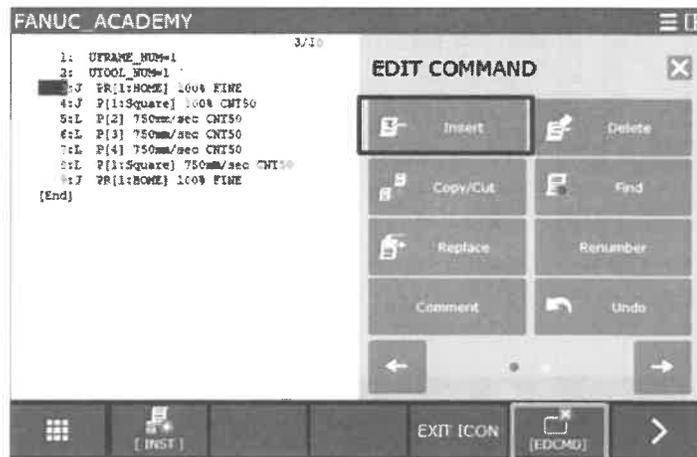
## Handling Tool

Adding blank lines to a program does nothing to affect cycle time as these lines are not executed by the processor. It is highly recommended to use this feature to break up your program to make easier to read.

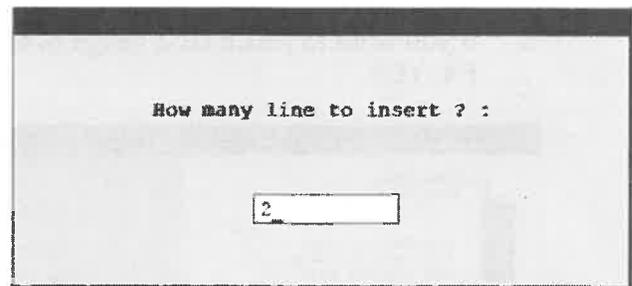
When you insert blank lines, you insert above the cursor position. It is possible to add just one line or multiple lines.

### Procedure 13-1 Inserting Blank Lines

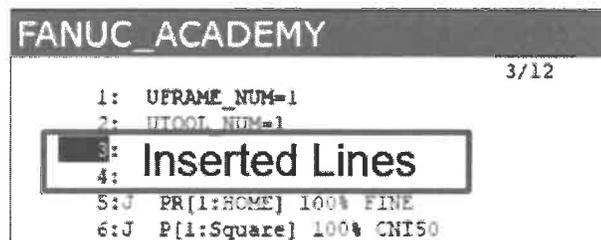
- 1 Place the cursor on the line below the spot to add blank line(s).
- 2 Press F5, [EDCMD]
  - a If F5 does not display [EDCMD] press NEXT.
- 3 Select INSERT



- 4 Enter the number of lines to add.
  - a Press ENTER.



**NOTE:** If you wish only add one line you can just press enter without entering the number one.



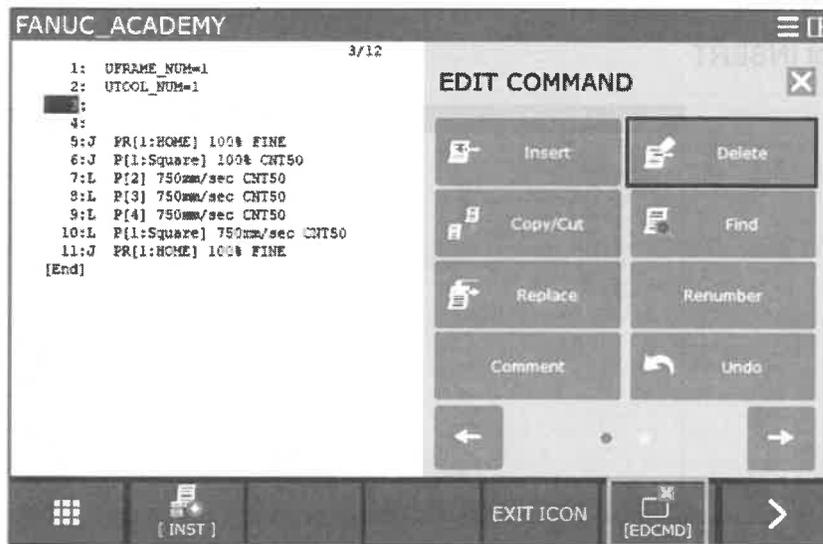
# 13: Edit Command

## 13.4 Deleting Lines

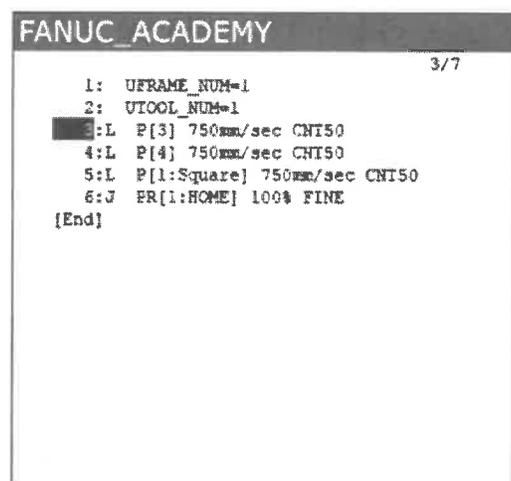
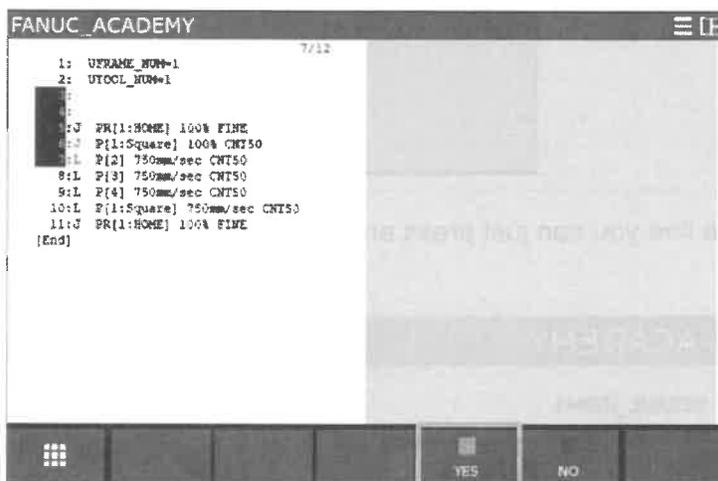
When you delete line from a program, you can either delete one line or a range of lines.

### Procedure 13-2 Deleting Line(s)

- 1 Place the cursor on the line or first line of the range that you want to delete.
- 2 Press F5, [EDCMD]
  - a If F5 does not display [EDCMD] press NEXT.
- 3 Select DELETE



- 4 If you want to DELETE just one line press F4, YES
- 5 If you want to DELETE a range of lines cursor up or down to select the range, then press F4, YES.



**13.5 Copy and Paste**

When you copy and paste, you select a group of instructions, make a copy of the group, and insert the group at one or more locations in the program.

**13.5.1 Pasting Options**

There are eight different ways to paste the lines that have been copied. Refer to the table below for a description of each type and an example of the pasted instructions using the lines provided below.

**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: CALL HOME
```

| Method        | Description                                                                                                                                                                                                         | Pasted Instructions                                                                                                                          |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| LOGIC (F2)    | <ul style="list-style-type: none"> <li>Pastes the lines exactly as they were copied</li> <li>Does not record positions</li> <li>Leaves the position numbers blank</li> </ul>                                        | <pre>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: CALL HOME</pre> |
| POS_ID (F3)   | <ul style="list-style-type: none"> <li>Pastes the lines exactly as they were copied</li> <li>Retains the original position numbers</li> </ul>                                                                       | <pre>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: CALL HOME</pre>         |
| POSITION (F4) | <ul style="list-style-type: none"> <li>Pastes the lines exactly as they were copied</li> <li>Renumbers the copied positions with next available position numbers</li> <li>Retains copied positional data</li> </ul> | <pre>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: CALL HOME</pre>         |
| CANCEL (F5)   | Cancels the paste and retains the copied lines so you can paste them elsewhere                                                                                                                                      |                                                                                                                                              |

# 13: Edit Command

| Method               | Description                                                                                                                                                                 | Pasted Instructions                                                                                                                              |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| R-LOGIC<br>(NEXT+F1) | <ul style="list-style-type: none"> <li>Pastes the lines in reverse order</li> <li>Does not record positions</li> <li>Leaves the position numbers blank</li> </ul>           | 7: CALL HOME<br>8: L P[...] 10mm/sec FINE<br>9: J P[...] 50% CNT50<br>10: DO[1] = ON<br>11: L P[...] 500mm/sec CNT80<br>12: J P[...] 100% CNT100 |
| R-POSID<br>(NEXT+F2) | <ul style="list-style-type: none"> <li>Pastes the lines in reverse order</li> <li>Retains the original position numbers</li> </ul>                                          | 7: CALL HOME<br>8: L P[4] 10mm/sec FINE<br>9: J P[3] 50% CNT50<br>10: DO[1] = ON<br>11: L P[2] 500mm/sec CNT80<br>12: J P[1] 100% CNT100         |
| R-POS<br>(NEXT+F4)   | <ul style="list-style-type: none"> <li>Pastes the instructions in reverse order</li> <li>Renumbers the copied positions with the next available position numbers</li> </ul> | 7: CALL HOME<br>8: L P[8] 10mm/sec FINE<br>9: J P[7] 50% CNT50<br>10: DO[1] = ON<br>11: L P[6] 500mm/sec CNT80<br>12: J P[5] 100% CNT100         |

### 13.5.2 Reverse Motion Pasting

Reverse motion pasting will redistribute the motion instructions that were pasted to maintain the path. If you remember from Chapter 10 and 11 when recording positions and executing a motion instruction everything is recorded at the destination.

If you were to copy and paste lines in reverse, as seen above, you are simply flipping the instructions upside down. This will result in a completely different path.

Reverse motion pasting will ensure that the path that followed to a specific spot will be maintained on the way out. A good example would be moving to a part to pick up. The path is defined on the way to the part. It stands to reason that following the exact path back out will be safe as well. Reverse motion pasting can accomplish this.

Refer to the table on the following page for a description of each type of reverse motion pasting.

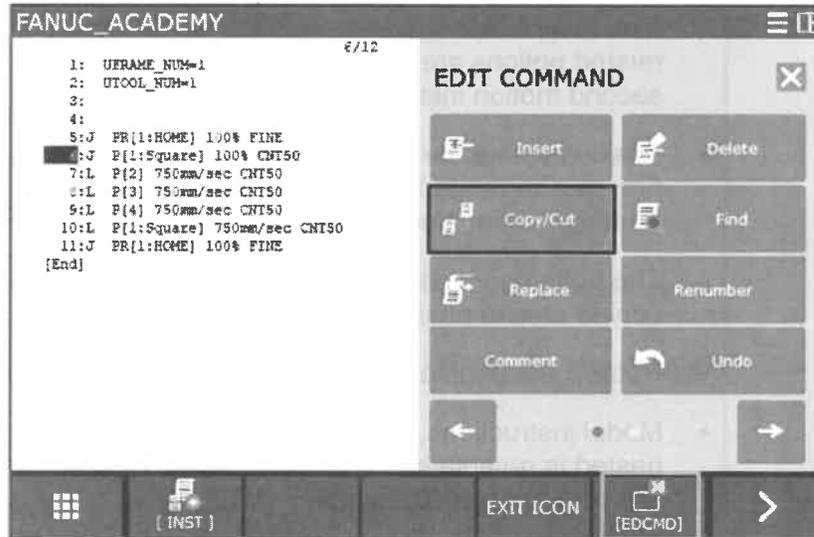
| Method                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Pasted Instructions                                                                                                                                                                                                              |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RM-POSID<br>(NEXT+F3) | <ul style="list-style-type: none"> <li>• Pastes all instructions except motion instructions in reverse order</li> <li>• Motion instructions are created using the current (first) and next (second) motion instruction:</li> <li>• Motion type, speed, and motion-related options are used from the second motion instruction</li> <li>• Positional data, termination type, and non-motion-related options are used from the first motion instruction</li> <li>• The bottom motion instruction in the copied area is pasted as is</li> <li>• Retains the original position numbers</li> <li>• 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</li> <li>• 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</li> </ul> | <b>1<sup>st</sup>:</b><br>J P[i] 100% CNT100,<br>Offset,PR[x]<br><b>2<sup>nd</sup>:</b><br>L P[j] 50 mm/sec FINE ACC150<br>PTH<br><b>Result:</b><br>L P[j] 50mm/sec FINE ACC150<br>L P[i] 50mm/sec CNT100<br>Offset,PR[x] ACC150 |
| RM-POS<br>(NEXT+F5)   | <ul style="list-style-type: none"> <li>• Same as RM-POSID except, renumbers the copied positions with the next available position numbers</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | <b>1<sup>st</sup>:</b><br>J P[k] 100% CNT100,<br>Offset,PR[x]<br><b>2<sup>nd</sup>:</b><br>L P[l] 50 mm/sec FINE ACC150<br><b>Result:</b><br>L P[l] 50mm/sec FINE ACC150<br>L P[k] 50mm/sec CNT100<br>Offset,PR[x] ACC150        |

# 13: Edit Command

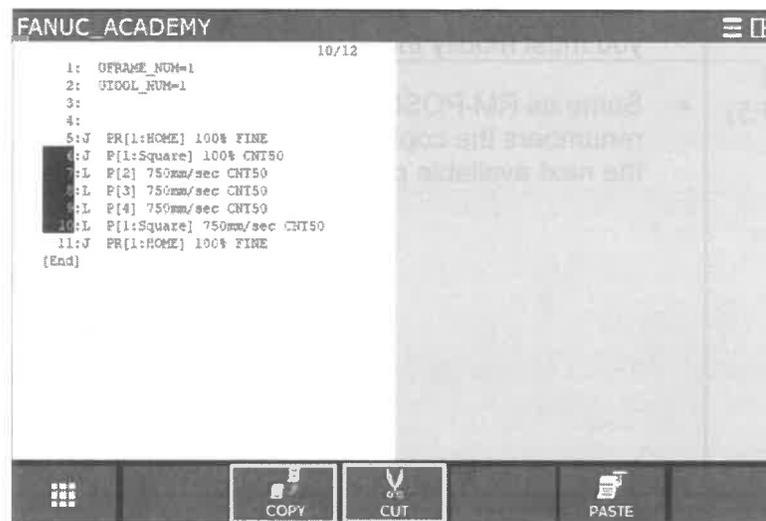
## 13.5.3 Execute a Copy and Paste Operation

### Procedure 13-3 Copy and Paste

- 1 Place the cursor on the line or first line of the range that you want to copy.
- 2 Press F5, [EDCMD]
  - a If F5 does not display [EDCMD] press NEXT.
- 3 Select COPY/CUT



- 4 With the cursor positioned on the line to copy or first line in the range to copy, press F2, SELECT
- 5 Move the cursor up or down to select the range to copy if necessary.
- 6 Press F2, COPY or F3, CUT.

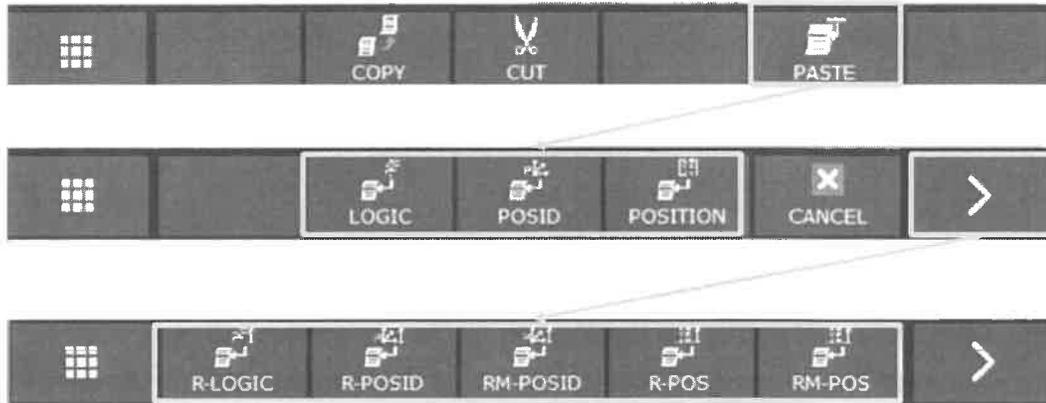


7 Navigate to the line below the location you wish to paste.

**NOTE:** Just like with INSERT, when you paste you paste above the cursor position.

8 Press F5, PASTE.

9 Select the appropriate paste method.



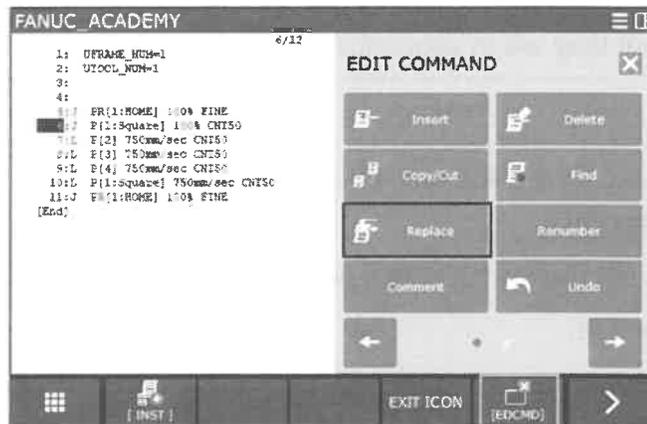
## 13.6 Replacing Instructions

When you perform a replace operation, you find specific instructions and replace those instructions with new instructions. This function is useful, for example, when setup information that affects the program is changed.

The replace operation searches from the current position of the cursor to the [END] command.

### Procedure 13-4 Performing a Replace Operation

- 1 Move the cursor to the line number of any instruction.
- 2 Press F 5, [EDCMD]
  - a If F 5 does not display [EDCMD] press NEXT.
- 3 Select Replace.



# 13: Edit Command

- 4 Select the instruction you want to replace from the list of instructions. Follow the information on the screen to specify the instruction



- 5 You will first be prompted to find an occurrence of the instruction.
- 6 You will then be prompted to enter the replacement information.
- 7 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.
- 8 Decide how to replace the item
  - a To replace the existing instruction with a new instruction in all instances, press F2, ALL. The system will prompt you to confirm that you want to replace all instances of the instruction.
  - b To replace the existing instruction with the new instruction, press F3, YES. The system will prompt you to search for the next one.
  - c To ignore this instance and find the next, press F4, NEXT, and the system will find the next instance, if there is one.
  - d To cancel the replace operation, press F5, EXIT.

## 13.7 Renumbering Positions

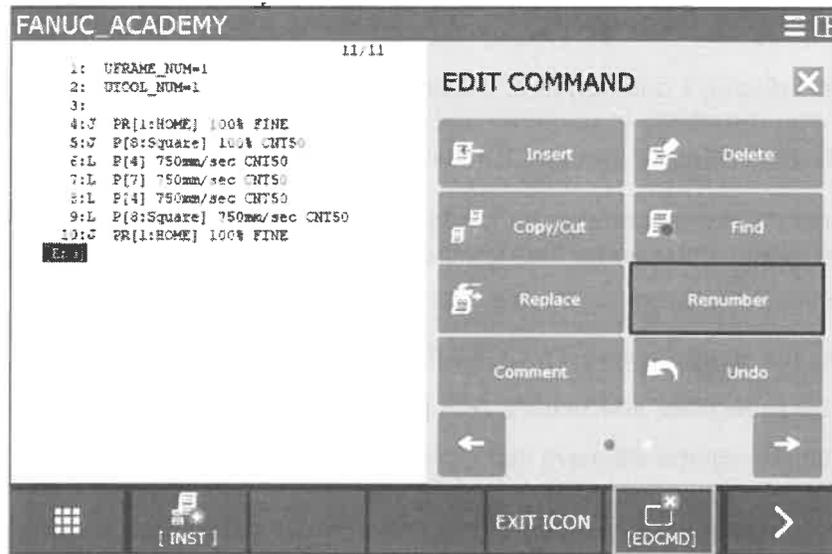
When you add positions in a program, the first available position number is assigned to the position, regardless of its place in the program. When you delete motion instructions, all remaining positions keep their current numbers. Renumbering reassigns all position numbers in the program so that they are in sequential order.

---

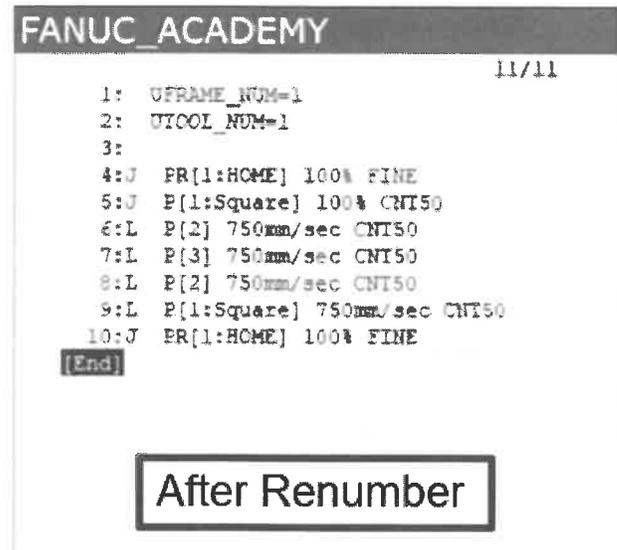
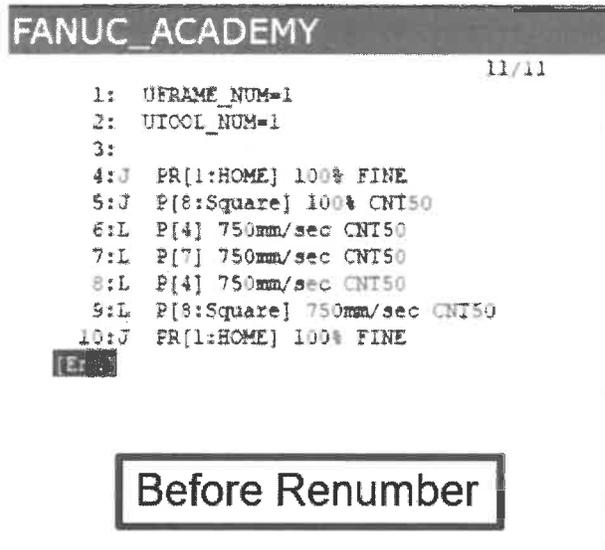
### Procedure 13-5 Renumbering Positions

---

- 1 Move the cursor to the line number of any instruction.
- 2 Press F5, [EDCMD]
  - a If F5 does not display [EDCMD] press NEXT.
- 3 Select Renumber.



- 4 Renumber the positions:
  - a If you do not want to renumber positions press F5, NO.
  - b To renumber positions press F4, YES.



**NOTE:** Like Position ID's will be alike ID after the renumber is performed.

**NOTE:** Renumbering has no effect on Position Register ID's.

## 13.8 Displaying Comments

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. Enabling and disabling comments is a global setting.

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])
- Register instructions (R [i:COMMENT])
- Position register instructions (PR [i:COMMENT])
- Position register element instructions (PR [i, j:COMMENT])

 **NOTE:** Comments associated with a Position (P [ ]) are stored within the program and cannot be disabled.

The display of the comments is turned on and off when you press F5, [EDCMD] and select [Comment].

```
FANUC_ACADEMY 11/11
1: UFRAME_NUM=1
2: UTOOL_NUM=1
3:
4:J PR[1:HOME] 100% FINE
5:J P[1:Square] 100% CNT50
6:L P[2] 750mm/sec CNT50
7:L P[3] 750mm/sec CNT50
8:L P[2] 750mm/sec CNT50
9:L P[1:Square] 750mm/sec CNT50
10:J PR[1:HOME] 100% FINE
[End]
```

**Comment Enabled**

```
FANUC_ACADEMY 11/11
1: UFRAME_NUM=1
2: UTOOL_NUM=1
3:
4:J PR[1] 100% FINE
5:J P[1:Square] 100% CNT50
6:L P[2] 750mm/sec CNT50
7:L P[3] 750mm/sec CNT50
8:L P[2] 750mm/sec CNT50
9:L P[1:Square] 750mm/sec CNT50
10:J PR[1] 100% FINE
[End]
```

**Comment Disabled**

## 13.9 Undoing Operations

You can undo an operation 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.

## Procedure 13-6 Undoing an Edit

- 1 Press F5, [EDCMD]
  - a If F5 does not display [EDCMD] press NEXT.
- 2 Select Undo.



- 3 Undo the operation. The recovered operation is displayed.
  - a If you do not want to undo the operation, press F5, NO.
  - b To undo the operation, press F4, YES.
- 4 To cancel the undo operation, press F5, [EDCMD] , select Undo. Press F4, YES, to redo the operation.

### 13.9.1 Undo Limitations

**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 keyboard
- The teach pendant ON/OFF switch is OFF
- The program is write-protected
- The teach pendant does not have enough available memory

**Also, you cannot Undo an operation in HandlingTool when any of the following conditions exist:**

- Modifying palletizing command
- Deleting program lines that include palletizing instructions.
- Copying program lines that include palletizing instructions.

# 13: Edit Command

- Pasting lines that include palletizing instruction.
- Replacing or re-numbering at the place in the program that has palletizing instructions.

**In addition, Undo will not work if you have executed any of the following instructions in a program:**

- Palletizing instructions (HandlingTool)
- Line tracking instructions
- On-line touch up

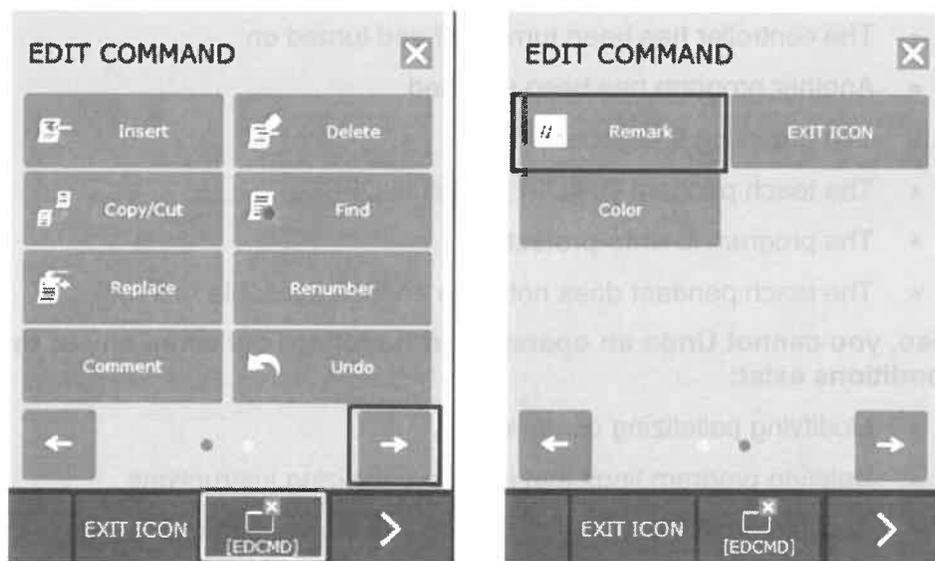
**NOTE:** If power fails in the process of performing an Undo operation, unexpected results can occur, and the desired modification is not guaranteed to have taken effect when power is restored.

## 13.10 Remarking Instructions

When you remark an instruction, you disable the instruction by commenting it out (“//” is shown at the beginning of the instruction). You cannot remark just a portion of the instruction. You will not be able to edit the instruction until you unremark it. You can perform other modifications on the instruction such as delete, copy and paste, find and replace, and renumber.

### Procedure 13-7 Remarking (disable) / Unmarking (enable) Instructions

- 1 Move the cursor to the line number of the instruction you want to remark or unremark. If you want to change several instructions in consecutive order, move the cursor to the first line to be changed.
- 2 Press F5, [EDCMD]
  - a If F5 does not display [EDCMD] press NEXT.
- 3 Select Remark.



# 13: Edit Command

## HandlingTool

- 4 To remark/unremark a range of lines, move the cursor to select the lines to be changed.
  - a The line number of each line to be changed will be highlighted as you move the cursor.
- 5 Remark or unremark the line or lines:
  - a To remark the selected line(s) press F4, REMARK
  - b To unremark the selected line(s) press F5, UNREMARK

```
FANUC_ACADEMY 11/11
1: UFRAME_NUM=1
2: UTOOL_NUM=1
3:
4: //J PR[1] 100% FINE
5: //J P[1:Square] 100% CNT50
6: //L P[2] 750mm/sec CNT50
7: //L P[3] 750mm/sec CNT50
8: //L P[2] 750mm/sec CNT50
9: //L P[1:Square] 750mm/sec CNT50
:
10: //J PR[1] 100% FINE
[End]
```

**Lines Remarked**

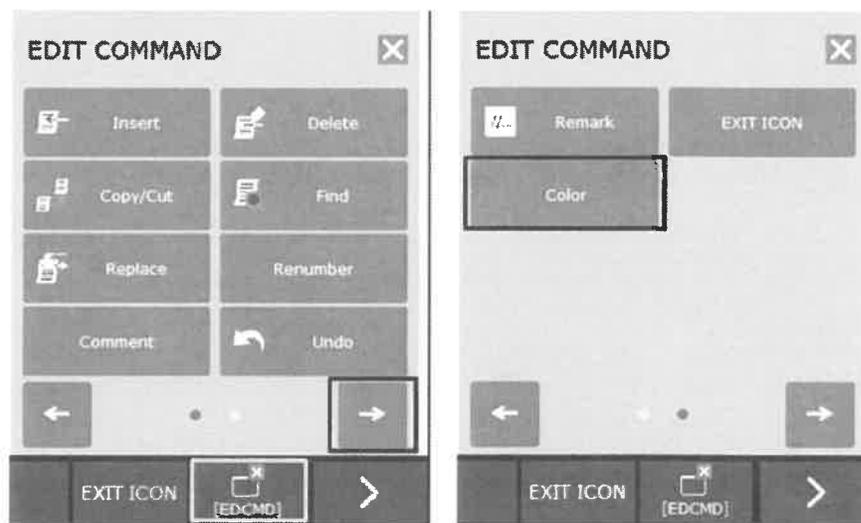
```
FANUC_ACADEMY 4/11
1: UFRAME_NUM=1
2: UTOOL_NUM=1
3:
4: J PR[1] 100% FINE
5: J P[1:Square] 100% CNT50
6: L P[2] 750mm/sec CNT50
7: L P[3] 750mm/sec CNT50
8: L P[2] 750mm/sec CNT50
9: L P[1:Square] 750mm/sec CNT50
10: J PR[1] 100% FINE
[End]
```

**Lines Unremarked**

### 13.11 Displaying Instructions in Color

The application instructions and the remark instructions can be displayed in color. The application instructions are displayed in red. The remark instructions are black on a yellow background.

The use of color can be turned on or off by pressing F5, [EDCMD] and selecting [Color].



## 13.12 Chapter Review

1. Which type of pasting will paste the motion instructions in order and renumber the Position ID's based on first available Position ID?
  - A LOGIC
  - B POSITION
  - C POS\_ID
  - D R\_POS
  
2. Which type of pasting will redistribute motion information so that the path is maintained in reverse and maintain Position ID's?
  - A R\_POS
  - B R\_POSID
  - C RM\_POS
  - D RM\_POSID
  
3. Which EDCMD feature would allow the programmer to change all FINE termination type to CNT termination type?
  - A COMMENT
  - B REPLACE
  - C UNDO
  - D FIND
  
4. If a line is remarked it can be edited?
  - A True
  - B False

## Lab 22 Utilize EDCMD

---

**Student Name:** \_\_\_\_\_

**Assignment:** The student will:

- Copy an entire program.
  - Copy individual program lines.
  - Delete program lines
  - Insert blank lines in a program
  - Replace motion variables
- 

**Condition:** Given a FANUC Robot and a controller

---

- Step:**
- 1** Copy the entire A\_SHAPES program and rename it A\_SQUARE. (See **Section 11.3.1 Select Screen Soft Keys, pg. 161**)
    - a** Delete all of the lines that are not part of the square. (See **Section 13.4 Deleting Lines, pg. 212**)
    - b** Insert a blank line between the frame instructions and the home position. (See **Section 13.3 Inserting Blank Lines, pg. 210**)
    - c** Utilizing REPLACE, change all Linear speeds from 750 mm/sec to 6000cm/min. (See **Section 13.6 Replacing Instructions, pg. 217**)
  - 2** Open the A\_SHAPES program, copy all of the lines that trace the circle. (See **Section 13.5.3 Execute a Copy and Paste Operation, pg. 216**)
    - a** Create a new program called A\_CIRCLE.
    - b** Paste the lines into A\_CIRCLE.
    - c** Copy the frame instructions from A\_SHAPES and paste them at the top of A\_CIRCLE.
    - d** Insert a blank line between the frame instructions and the home position.
    - e** Utilizing REPLACE, change all CNT termination to FINE termination.
    - f** Copy only the instructions that trace the circle and paste them in reverse.
  - 3** Open the A\_SHAPES program, copy all of the lines that trace the triangle.
    - a** Create a new program called A\_TRIANGLE.
    - b** Paste the lines into A\_TRIANGLE
    - c** Copy the frame instructions from A\_SHAPES and paste them at the top of A\_TRIANGLE.

# 13: Edit Command

HandlingTool

- d Insert a blank line between the frame instructions and the home position.
  - e Utilizing REPLACE, change all CNT 75 termination to CNT 50 termination.
  - f Utilizing REPLACE, change all Linear motion speed from 500mm/sec to 1000mm/sec.
- 4 Test all of the newly created programs

Completed:

Instructor: \_\_\_\_\_

## 14 INPUTS AND OUTPUTS (I/O)

### 14.1 Chapter Objectives

- I/O Overview
- Types of I/O
- Types of I/O Hardware
- Configuring I/O
- Configure Group I/O
- Monitor and Force I/O

### 14.2 I/O Overview

Inputs and outputs (I/O) are electrical signals that enable the controller to communicate with the robot, end-of-arm tooling, and other external devices such as a PLC (Programmable Logic Controller, HMI (Human Machine Interface), sensors and actuators. When properly programmed and properly setup these signals allow for communication between the robot controller and other devices. This is similar to how humans speak (output) and listen (input), but for machines the signals are electrical.

In modern manufacturing there are many ways to interface devices and FANUC America Corporation offers many solutions depending on a user's preferred I/O communication method or application. This chapter is intended to be an introduction to some of these communication options.

- The controller core software views I/O as signals
  - Outputs can be turned off/on, or set to a value if supported.
  - Inputs can be read, their values stored, and decisions made.
- I/O Signals are identified by kind and number, for example:
  - DI [1] = DIGITAL INPUT 1
  - GO [2] = GROUP OUTPUT 2

### 14.3 Types of I/O

The following kinds of I/O are available. The amounts and kinds vary depending on the kind of I/O hardware you use:

- Digital I/O
- Analog I/O
- Standard Operator Panel I/O
- Group I/O
- User Operator Panel (UOP) I/O
- Robot I/O

# 14: Inputs and Outputs

## 14.3.1 Digital

A digital I/O signal (DI or DO) is a control signal sent to or from the controller. Digital signals can have only one of two possible states: ON or OFF. Digital signals provide access to data on a single input or output signal line.

## 14.3.2 Group

Group I/O signals (GI or GO) are made up of a sequence of digital I/O signals. These are interpreted as a binary integer.

A group signal is a group of up to 16 ones (1) and zeros (0), indicated ON or OFF. Each bit in group input or output is a single digital input or output line. Unused bits are assigned a "0."

If you group your digital I/O signals, you can control or monitor the sequence of I/O signals within your program by setting or reading a single group using the group I/O instructions.

## 14.3.3 Analog

An analog I/O signal (AI or AO) is an input or output voltage that has a value within the range of the I/O board or module used.

Analog input devices convert external analog signals into numeric signals for use by the controller. Analog output devices allow analog signals to be sent out to external devices. Typical values of analog inputs and outputs are from -10 volts to 0 volts, or from 0 volts to +10 volts. These values depend on the particular device used in your controller.

## 14.3.4 User Operator Panel I/O

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

Most UOP I/O signals (UI or UO) are active when the robot is in a remote condition. Signals that affect safety are always active.

For systems with a process I/O board, the UOP signals are configured by default to dedicated ports.

## 14.3.5 Standard Operator Panel (SOP) I/O

SOP I/O signals (SI or SO) correspond to internal controller software panel digital input and output signals that control the operator panel on the controller. You cannot change SOP I/O assignments, but might find it to be useful to display them during troubleshooting.

## 14.3.6 Robot I/O

Robot I/O signals (RI or RO) consist of the input and output signals between the controller and the robot. These signals are sent to the EE (End Effector) connector located on the robot. The number of robot input and output signals may vary based on application software.

- Handling Tool Application software has eight Robot Inputs and eight Robot Outputs.
- LR Handling Tool Application software has up to six Robot Inputs and six Robot Outputs

# 14: Inputs and Outputs

## 14.4 I/O Hardware

The type of I/O hardware that is used will dictate the type of I/O available as well as the number of ports.

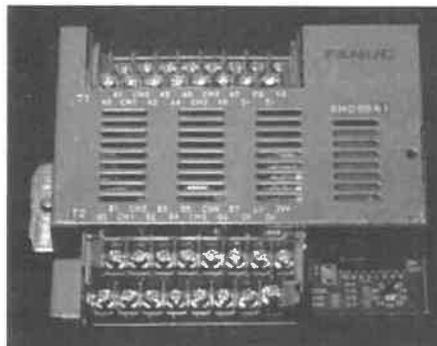
### 14.4.1 Model A I/O

- Discrete I/O modules
- Communicates with CPU through interface module
- Requires backplane in controller
- Digital in/out
  - 8/16/32 inputs
- Analog in/out
  - 2/4 channels



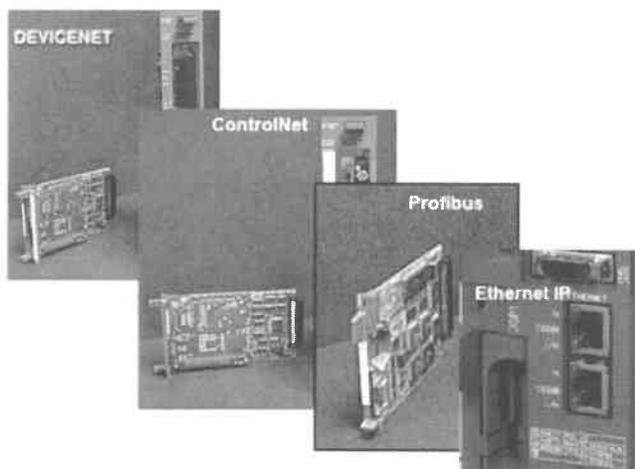
### 14.4.2 Model B I/O

- Discrete I/O modules
  - Can have both inputs and outputs on same unit
- Digital Only
- 24v DC or 120v AC
- Serial communications with CPU via interface module
- No backplane required



### 14.4.3 Other Types

- DeviceNet
- ControlNet (legacy)
- Profibus
- Profinet
- Ethernet IP
- EtherCat



**NOTE:** These types of I/O hardware will require software options and possibly additional hardware to configure.

## 14.5 Setting Up I/O (Configuring)

When all appropriate I/O hardware has been installed and connected, you must configure the I/O. Configuring I/O establishes the correspondence between the signal number and the physical port. Each signal, or signal-sequence must be configured to a **rack**, a **slot in the rack**, and the **starting point or channel number (analog)**. You can change this configuration depending on the kind of I/O you are using. Model "A" I/O is unique, in the fact that some FANUC America Corporation software will be automatically configured, similar to the PC-world's "Plug and Play".

 **NOTE:** After configuring I/O or changing I/O configuration, you must turn OFF the controller and then turn it back ON to make the change take effect.

### 14.5.1 Rack

Rack assignment lets the robot software know the type of I/O communication that will be used. To simplify things, the Rack assignment is similar to how humans communicate using different forms such as talking, e-mail, texting, paper notes, etc. For a robot the Rack assignment lets the software know how the communication will be done.

The rack is the first part of the address for an I/O signal, it is the physical location where the module or board is mounted

**The following ground rules apply to assigning I/O rack numbers:**

- Model A or Model B I/O Starts at rack 1
  - Racks are numbered sequentially
- Devicenet uses rack 81-84
- EtherNet uses rack 89
- Profibus uses rack 66/67
- FL-net uses rack 68
- PMC uses rack 33
- Mate Controller uses rack 48

### 14.5.2 Slot

The slot is the 2<sup>nd</sup> part of the address for an I/O signal. The slot number distinguishes individual I/O modules on a rack. Identifies space on the rack where the individual module is connected.

**The following rules apply to slot assignment:**

- Valid numbers are 1 through 10, no letters
- Slot numbers are assigned sequentially
- The first process I/O board is always assigned slot 1
- Slot numbers cannot be used twice in the same rack

# 14: Inputs and Outputs

## 14.5.3 Starting Point (Digital) / Channel (Analog)

The last part of the assignment is the Starting Point (Digital I/O) or the Channel (Analog I/O). This is the physical position on the I/O module or process I/O board that identifies the first port in a range

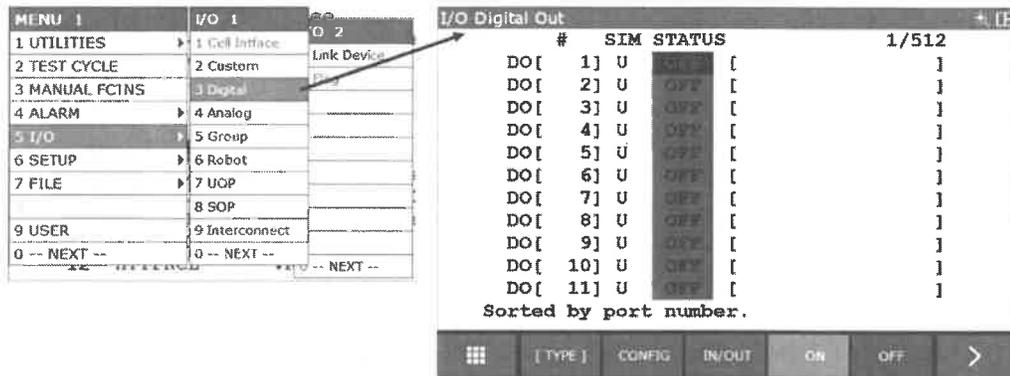
## 14.5.4 Configuring I/O

As discussed above, to configure I/O you must have the Rack, Slot, and Start Point for the range to set up. This section will walk you through how to configure Digital I/O using a FANUC model A I/O rack.

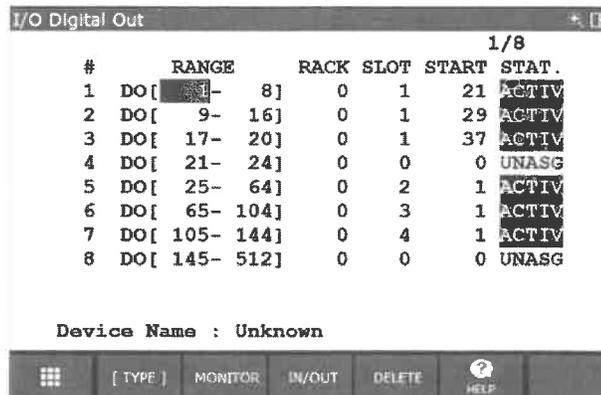
### Procedure 14-1 Configuring Digital I/O

**NOTE:** Analog and digital I/O is configured automatically by the system at Cold start when the system variable \$IO\_AUTO\_CFG is set to TRUE. Use this procedure only if you want to change the configuration

- 1 Press MENU.
- 2 Highlight I/O.
  - a On the flyout menu cursor down to Digital.
  - b Press ENTER.

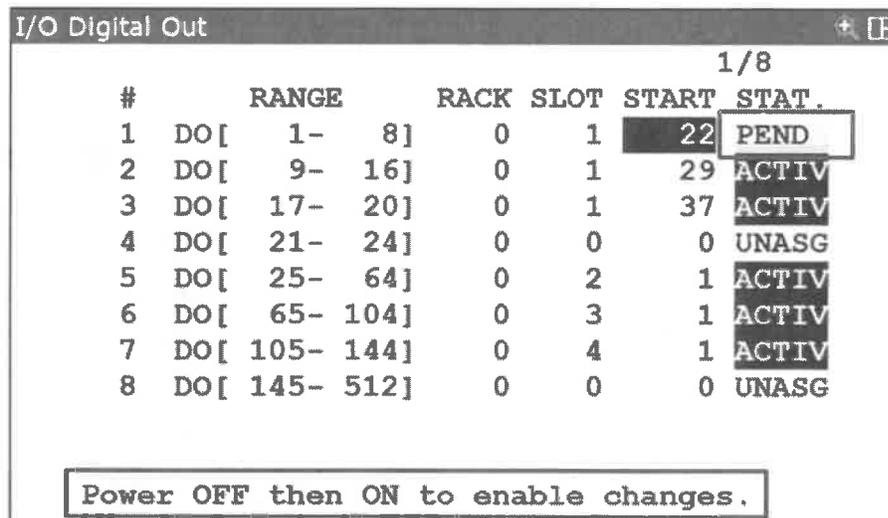


- 3 To configure digital signals press F2, CONFIG



# 14: Inputs and Outputs

- 4 You can toggle between Inputs and Outputs by pressing F3, IN/OUT.
- 5 Configure the I/O
  - a Set the range of the assignment.
  - b Move the cursor to RACK, type the value, and press ENTER.
  - c Move the cursor to SLOT, type the value, and press ENTER.
  - d Move the cursor to START, type the value, and press ENTER.
  - e To delete the assignment for a range of ports, either set the RACK, SLOT, and START values to zero, or press F4, DELETE, and then confirm the deletion. After a deletion, the range of ports will be merged with the UNASG lines immediately above or below it automatically.
- 6 Verify the assignment is valid.
  - a A valid assignment will list as PEND (pending)
  - b Cycle power to enable changes.



| # | RANGE         | RACK | SLOT | START | STAT. |
|---|---------------|------|------|-------|-------|
| 1 | DO[ 1- 8]     | 0    | 1    | 22    | PEND  |
| 2 | DO[ 9- 16]    | 0    | 1    | 29    | ACTIV |
| 3 | DO[ 17- 20]   | 0    | 1    | 37    | ACTIV |
| 4 | DO[ 21- 24]   | 0    | 0    | 0     | UNASG |
| 5 | DO[ 25- 64]   | 0    | 2    | 1     | ACTIV |
| 6 | DO[ 65- 104]  | 0    | 3    | 1     | ACTIV |
| 7 | DO[ 105- 144] | 0    | 4    | 1     | ACTIV |
| 8 | DO[ 145- 512] | 0    | 0    | 0     | UNASG |

Power OFF then ON to enable changes.

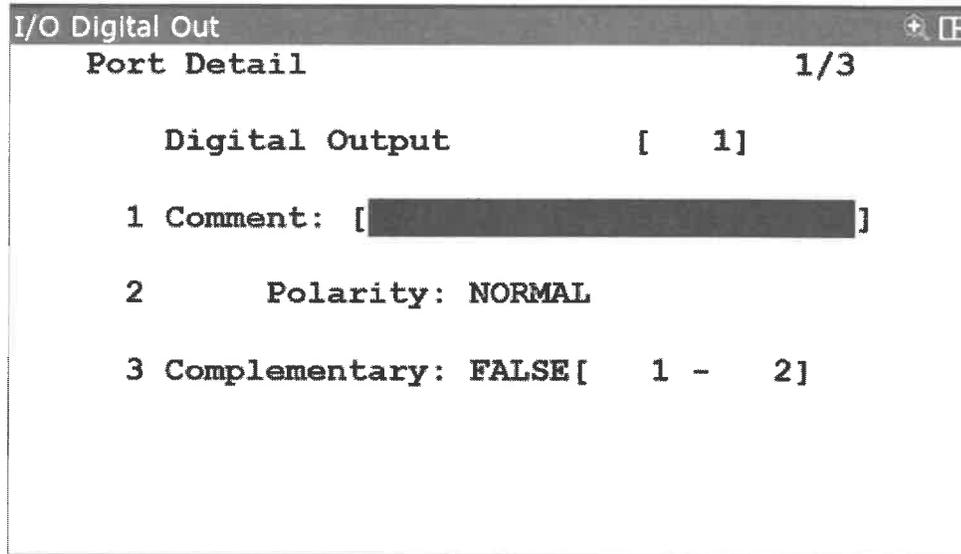
### 14.5.5 Digital I/O Comments, Complimenting, and Polarity

With the I/O properly configured the signals can be commented, complemented, and polarity changed. The use for these items will be directly related to the intended use within the automation system. All of these items are set at the signals DETAIL screen.

#### Procedure 14-2 Commenting, Complimenting, and/or Changing Polarity

- 1 Press MENU.
- 2 Highlight I/O.
  - a On the flyout menu cursor down to Digital.
  - b Press ENTER.
- 3 Cursor to the required signal.

- 4 Press NEXT
- 5 Press F4, DETAIL



- 6 To add a comment place the cursor on the comment line and press enter.
    - a Type in the name of the signal.
    - b Press ENTER.
  - 7 To change Polarity cursor to polarity.
    - a For Inverse Polarity press F4, INVERSE
    - b For Normal Polarity press F5, Normal
-  **NOTE:** Polarity is NORMAL by default.
- c Cycle power for this setting to take effect.
- 8 To make a complimentary pair cursor to Complementary.
    - a The numbers in the bracket define the two signals that will be complimented.
-  **NOTE:** You can only set an odd output complimentary with the corresponding even output.
- b Press F4, TRUE to compliment the pair
  - c Press F5, FALSE to uncompliment the pair
-  **NOTE:** All signal are uncomplimented by default.
- d Cycle power for this setting to take effect.

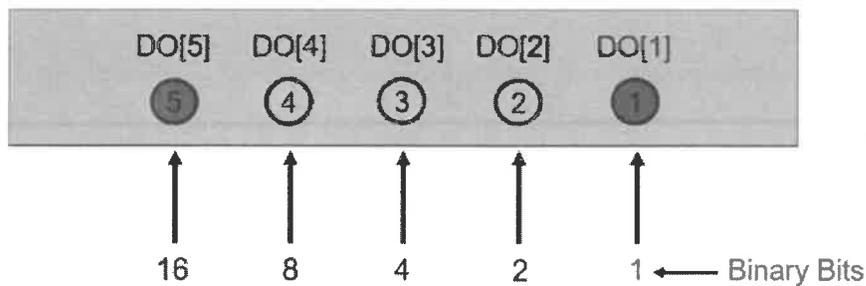
# 14: Inputs and Outputs

## 14.6 Group I/O

Group Inputs and Outputs are very helpful when programming your robot. Group I/O allows the user to “Group” Digital Inputs or Digital Outputs to help the robot understand the signals as an integer (aka: number).

Group Inputs are a user defined “team” of Digital Inputs used to provide the robot a numeric value. This is useful for program selection from a PLC for example. Likewise, Group Outputs allow the programmer to turn ON any combination of Digital Outputs with one easy program instruction using a numeric value.

Group I/O’s numeric value is based on a series of binary bits which sequence as 1, 2, 4, 8, 16, 32, 64, 128, etc. This number will represent which Inputs are being received or which Outputs are ON. The example shown below uses a GO (Group Output) value of 17 to turn on DO[1] and DO[5] based on their binary bit value. Group Outputs are also useful to turn OFF any Outputs which may have accidentally been left ON. (Example program instruction: GO[1] = 0)



### GROUPED INPUT

| DI[6] | DI[5] | DI[4] | DI[3] | DI[2] | DI[1] | Binary Count    | Integer Value |
|-------|-------|-------|-------|-------|-------|-----------------|---------------|
| 32    | 16    | 8     | 4     | 2     | 1     |                 |               |
| 0     | 0     | 0     | 0     | 0     | 0     | 0 =             | 0             |
| 0     | 0     | 0     | 0     | 0     | 1     | 1 =             | 1             |
| 0     | 0     | 0     | 0     | 1     | 0     | 2 + 0 =         | 2             |
| 0     | 0     | 0     | 0     | 1     | 1     | 2 + 1 =         | 3             |
| 0     | 0     | 0     | 1     | 0     | 0     | 4 + 0 + 0 =     | 4             |
| 0     | 0     | 0     | 1     | 0     | 1     | 4 + 0 + 1 =     | 5             |
| 0     | 0     | 0     | 1     | 1     | 0     | 4 + 2 + 0 =     | 6             |
| 0     | 0     | 0     | 1     | 1     | 1     | 4 + 2 + 1 =     | 7             |
| 0     | 0     | 1     | 0     | 0     | 0     | 8 + 0 + 0 + 0 = | 8             |
| 0     | 0     | 1     | 0     | 0     | 1     | 8 + 0 + 0 + 1 = | 9             |
| 0     | 0     | 1     | 0     | 1     | 0     | 8 + 0 + 2 + 0 = | 10            |
| 0     | 0     | 1     | 0     | 1     | 1     | 8 + 0 + 2 + 1 = | 11            |
| 0     | 0     | 1     | 1     | 0     | 0     | 8 + 4 + 0 + 0 = | 12            |
| 0     | 0     | 1     | 1     | 0     | 1     | 8 + 4 + 0 + 1 = | 13            |
| 0     | 0     | 1     | 1     | 1     | 0     | 8 + 4 + 2 + 0 = | 14            |
| 0     | 0     | 1     | 1     | 1     | 1     | 8 + 4 + 2 + 1 = | 15            |

**INPUT STATUS**  
 1 =ON  
 0 =OFF

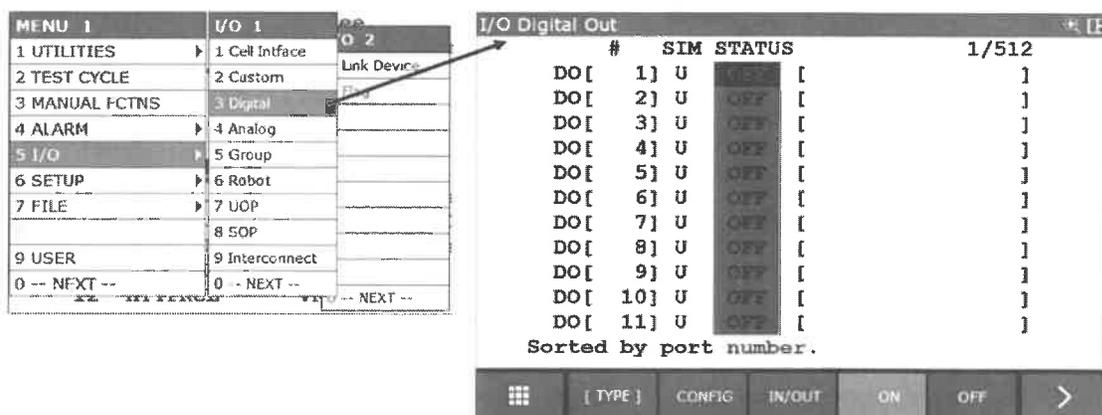
# 14: Inputs and Outputs

## 14.6.1 Configuring Group I/O

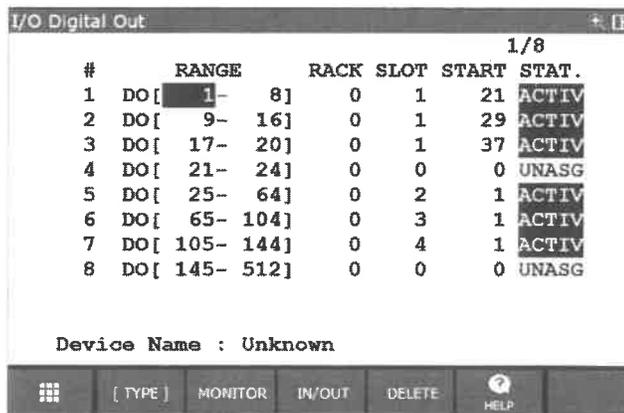
Configuring Group I/O is the same as configuring Digital I/O. The programmer must tell the controller which Digital I/O will be part of the group. This means providing the robot the Rack, Slot, Starting point, and Number of points for the I/O used in the Group (not to exceed 16).

### Procedure 14-3 Configuring Group I/O

- 1 Press MENU.
- 2 Highlight I/O.
  - a On the flyout menu cursor down to Digital.
  - b Press ENTER.



- 3 Display either the Digital Outputs or Digital Inputs by pressing F3, IN/OUT
- 4 With the correct type of Digital Signal displayed press F2, CONFIG

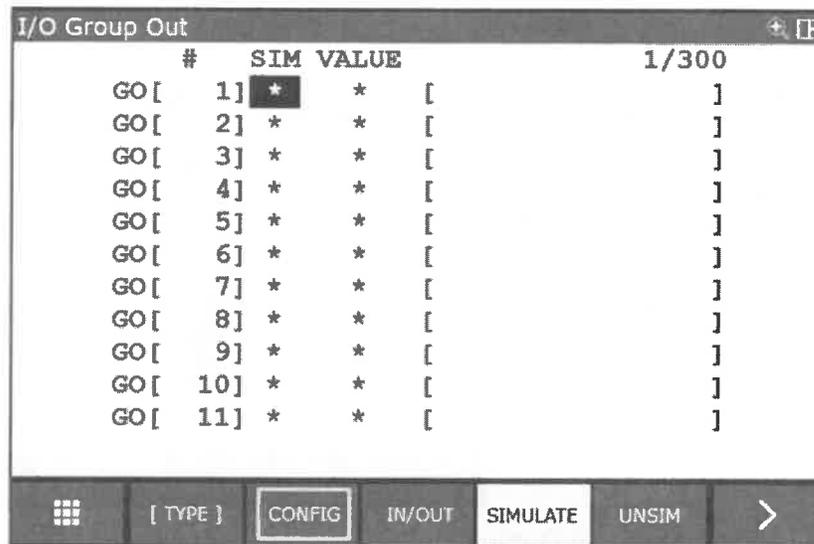


- 5 Determine the Digital signals that you intend to group.
  - a Find the Rack, Slot, and Start point for the first signal in the group.
  - b Determine how many digital signals will be in the group.

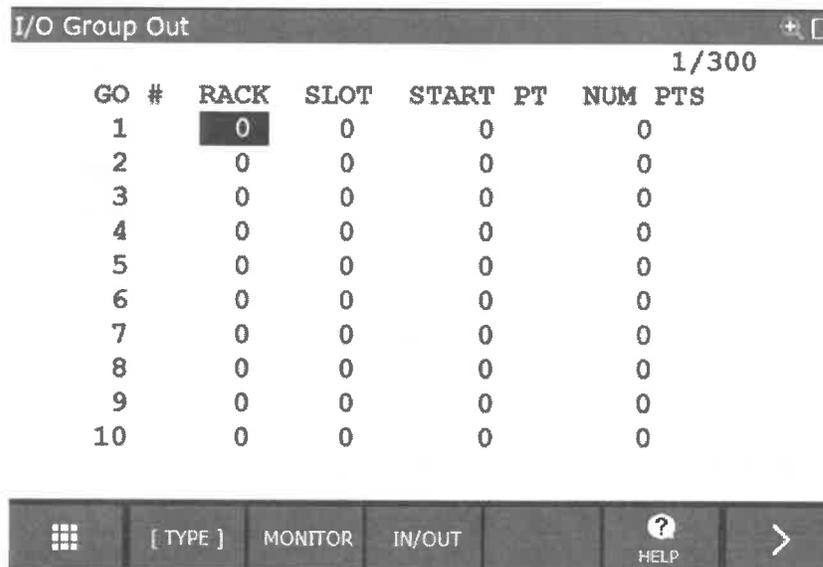
**NOTE:** The group must contain 2-16 signals, be sequential, and on the **same** rack and slot.

# 14: Inputs and Outputs

- 6 For example, these procedures will group DO [2-6]. Using the image after STEP 4 we can determine the configuration data needed.
    - a The data needed for the group is as follows: Rack: 0, Slot: 1, Start Point: 22, Number of Points: 5.
-  **NOTE:** These values are for demonstration purposes only, your configuration data may be different.
- 7 With the configuration data determined, press F1, [TYPE]
    - a Cursor to Group and press ENTER, or press the number on the keypad corresponding with GROUP I/O.



- 8 From the Group Output screen press F2, CONFIG
  - a If you are configuring Group Inputs press F3, IN/OUT to toggle to Group Input.



# 14: Inputs and Outputs

- 9 Enter the Rack, Slot, Start Point, and Number of Points that was determined from the Digital I/O screen.

| I/O Group Out |      |      |          |         |       |
|---------------|------|------|----------|---------|-------|
| GO #          | RACK | SLOT | START PT | NUM PTS |       |
| 1             | 0    | 1    | 22       | 5       | 1/300 |
| 2             | 0    | 0    | 0        | 0       |       |

- 10 With the values entered verify the assignment.
- Press NEXT.
  - Press F2, VERIFY.
  - A correct assignment will show, "Port assignment is valid".
  - An incorrect assignment will show, "Port assignment is invalid".

**NOTE:** If your assignment is invalid, verify the configuration data is correct and was entered correctly.

| I/O Group Out |      |      |          |         |       |
|---------------|------|------|----------|---------|-------|
| GO #          | RACK | SLOT | START PT | NUM PTS |       |
| 1             | 0    | 1    | 22       | 5       | 1/300 |
| 2             | 0    | 0    | 0        | 0       |       |
| 3             | 0    | 0    | 0        | 0       |       |
| 4             | 0    | 0    | 0        | 0       |       |
| 5             | 0    | 0    | 0        | 0       |       |
| 6             | 0    | 0    | 0        | 0       |       |
| 7             | 0    | 0    | 0        | 0       |       |
| 8             | 0    | 0    | 0        | 0       |       |
| 9             | 0    | 0    | 0        | 0       |       |
| 10            | 0    | 0    | 0        | 0       |       |

Port assignment is valid.

[ TYPE ] VERIFY >

- 11 With a valid assignment made, cycle power to enable changes.

## 14.7 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:

- Monitoring
- Forcing
- Simulating

# 14: Inputs and Outputs

## 14.7.1 Monitoring I/O

Monitoring I/O is simply viewing the status of the specific I/O on the iPendant. As the signals change the user can see the state change to ensure the signals are operating properly.

- To monitor I/O simply navigate to the I/O menu and select the type of I/O to monitor.
- You can switch between Outputs or Inputs with F3, IN/OUT.
- You can sort by comment or signal number by pressing NEXT, and either F2, NUM-SRT or F3, CMT-SRT.

## 14.7.2 Forcing I/O

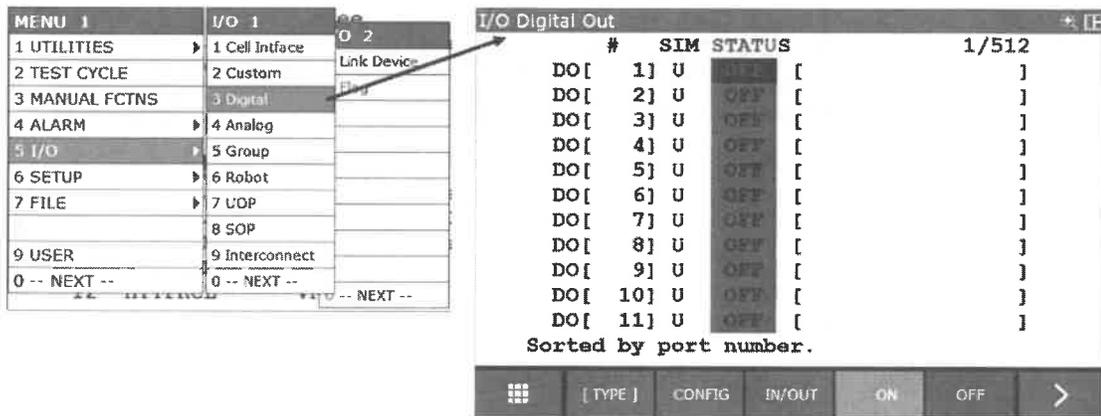
Forcing outputs is turning output signals on or off. Outputs can be forced in two ways; from the I/O screen or forced within a program using I/O instructions. Only outputs can be forced because these are signals sent by the controller, if you need to force an Input it must first be simulated (see next section).

### 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 you force it; otherwise, you could injure personnel or damage equipment.**

### Procedure 14-4 Forcing a Digital Output from the I/O Screen

- 1 Press MENU.
- 2 Highlight I/O.
  - a On the flyout menu cursor down to Digital.
  - b Press ENTER.



- 3 Display the Digital Outputs.
  - a If Digital Outputs are not displayed press F3, IN/OUT
- 4 Move the cursor to the STATUS of the output you want to force.

# 14: Inputs and Outputs

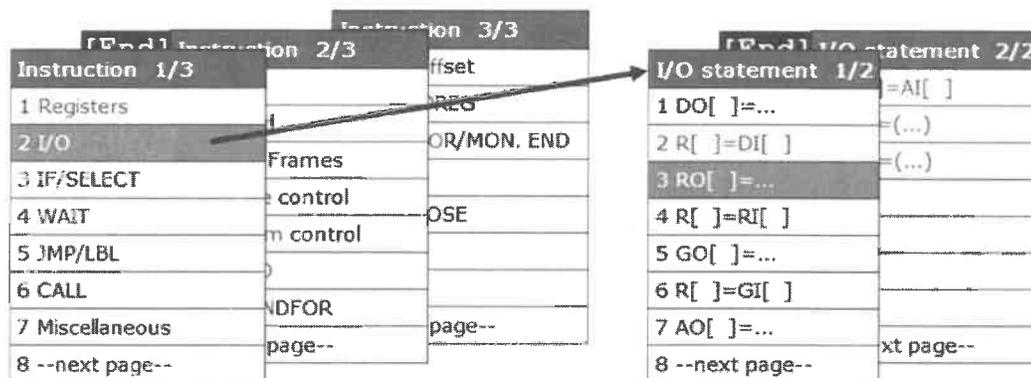
- 5 Press the function key that corresponds to the value you want.
  - a Press F4, ON
  - b Press F5, OFF

**NOTE:** For forcing a group output navigate to the Group I/O screen, enter the group value in the value column. Verify at the Digital Output screen the correct Outputs turned on.

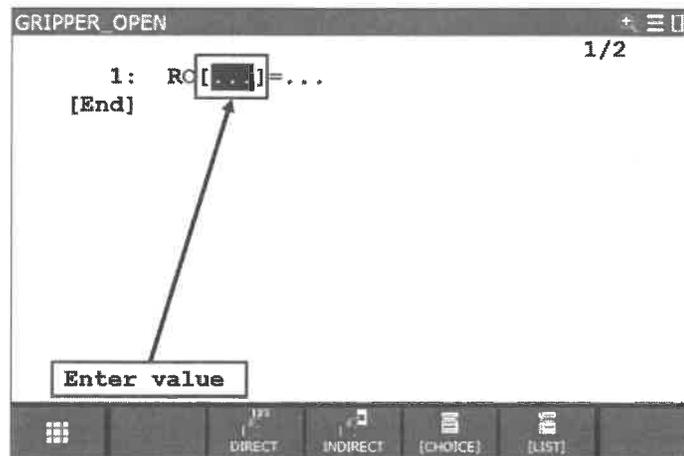
## Procedure 14-5 Forcing Outputs using Program Instructions

- 1 Place your cursor on the blank line in a program you intend to add the instruction.
- 2 Press F1, [INST]
  - a If F1 does not display [INST] press NEXT.
- 3 On the instruction menu cursor down to I/O or press the corresponding number on the key pad.
- 4 Choose the desired instruction.

**NOTE:** This demonstration will program the gripper to open using RO [1].

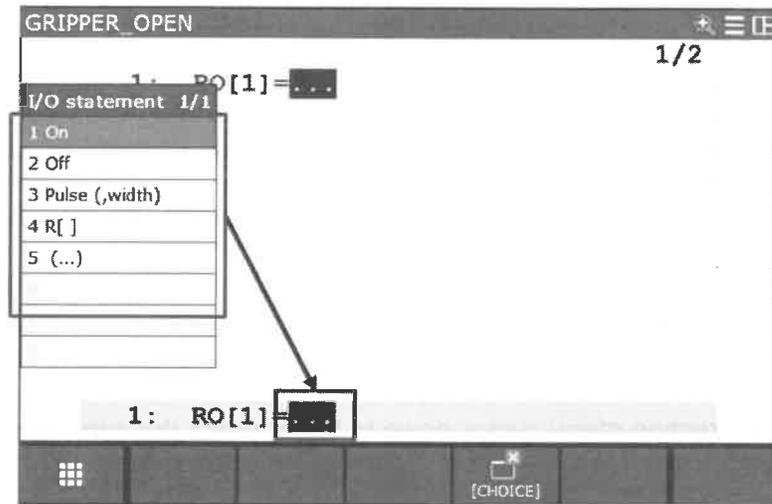


- 5 With the desired instruction chosen the software will prompt you to fill in the needed information to complete the instruction.



# 14: Inputs and Outputs

- 6 With the signal number identified, enter the desired status.



**NOTE:** On the training robots, the output to open and close the gripper must be complemented for the above program to open the gripper. The RO signals to open the gripper may not be RO [1].

**NOTE:** To program a Digital Output or Group Output into a program the steps are the same as above, only you specify DO or GO when you choose the instruction.

### 14.7.3 Simulating I/O

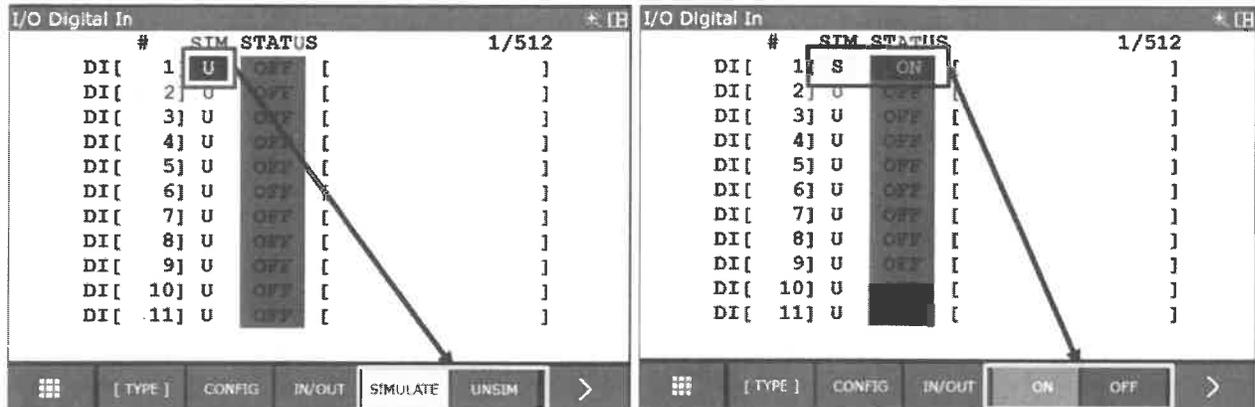
Simulating inputs and outputs is forcing inputs and outputs without signals entering or leaving the controller. This can be used to test program logic and motion when I/O devices and signals are not set up. You can simulate digital, analog, group, and robot I/O only; you cannot simulate UOP or SOP I/O. When you are finished simulating a signal you can reset, or unsimulate it.

#### Procedure 14-6 Simulating I/O

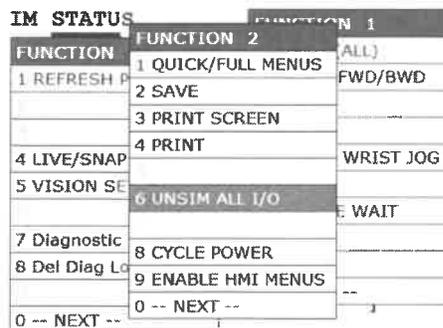
- 1 Press MENU.
- 2 Highlight I/O.
  - a On the flyout menu cursor down to type of I/O to simulate.
  - b Press ENTER.
- 3 If you simulate a signal, you can force the status by setting it to a value. When the signal is unsimulated, its actual status is displayed.

# 14: Inputs and Outputs

- 4 Move the cursor to the SIM column of the signal you want to simulate.
  - a U means the signal is not simulated or unsimulated.
  - b S means the signal is simulated.
- 5 Simulate or unsimulate the signal.
  - a To simulate, press F4, SIMULATE.
  - b To unsimulate, press F5, UNSIM.



- 6 To unsimulate all simulated signals, press FCTN and then select UNSIM ALL I/O.



## 14.8 Saving I/O Configuration Data

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

### Procedure 14-7 Saving I/O Settings

- 1 Set the Default Device, see Section 7.4.1 Setting the Default Device
  - a Create a subdirectory if necessary, see Section 7.4.3 Subdirectories
- 2 Navigate to the Digital I/O Screen
- 3 Press FCTN, -0- NEXT, SAVE
- 4 The file DIOCFGSV.IO will be saved.

# 14: Inputs and Outputs

## 14.9 Chapter Review

1. To configure group I/O all of the signals must be on the same rack and \_\_\_\_\_?
  - A Start Point
  - B Slot
  - C Hardware
  - D Analog
2. What type of I/O handles communication between the controller and the robot?
  - A Digital I/O
  - B UOP I/O
  - C Robot I/O
  - D SOP I/O
3. Before forcing an input it must be \_\_\_\_\_?
  - A Simulated
  - B Configured
  - C Part of a Group
  - D Inputs can never be forced
4. The item UNSIM ALL I/O can be found at which menu?
  - A I/O Menu
  - B Main Menu
  - C Function Menu
  - D Cell Setup Menu

# 14: Inputs and Outputs

## Lab 23

### Create a Group Output

**Student Name:** \_\_\_\_\_

**Assignment:** The student will:

- Configure a Group Output

**Condition:** Given a FANUC Robot and a controller

- Step:**
- 1 Determine the Rack, Slot, Start Point, and Number of Points to create a group output of DO [3-9]. (See Section 14.5.4 Configuring I/O, pg. 231)
    - a Rack: \_\_\_\_\_
    - b Slot: \_\_\_\_\_
    - c Start Point: \_\_\_\_\_
    - d Number of Points: \_\_\_\_\_
  - 2 Enter the data into the Group Output screen to configure GO [1]. (See Section 14.6.1 Configuring Group I/O, pg. 235)
  - 3 Force GO [1] to a value of 97. (See Section 14.6.1 Configuring Group I/O, pg. 235)
  - 4 Go to the Digital Output screen and verify the DO signals that are on.
  - 5 In the image below darken in the outputs that are on and note their bit value on the line provided.
  - 6 Do the outputs represent the correct forced value? \_\_\_\_\_.



=

- 7 Save your new I/O configuration to the Individual Files directory of the default device. (See Section 14.8 Saving I/O Configuration Data, pg. 241)

Completed:

Instructor: \_\_\_\_\_

## Lab 24

### Create an I/O Program

---

Student Name: \_\_\_\_\_

Assignment: The student will:

- Create a Program to Open the Gripper.
  - Create a Program to Close the Gripper.
- 

Condition: Given a FANUC Robot and a controller

---

- Step:**
- 1 Ensure the Robot Outputs that control the gripper are complimented. (See Section 14.5.5 Digital I/O Comments, Complimenting, and Polarity, pg. 232)
    - a Ask your instructor which signals control the gripper.
  - 2 Create a new program called A\_GRIP\_OPEN.
  - 3 Go to the program detail and make this a non-motion program, meaning disable motion group one. (See Section 11.6.5 Group Mask, pg. 168)
  - 4 Add the instruction: RO [ x ] = ON. (See Section 14.7.2 Forcing I/O, pg. 238)  
 **NOTE:** [x] is the signal number that opens the gripper.
  - 5 Create a new program called A\_GRIP\_CLOSE.  
**HINT:** Use the copy feature.
  - 6 Go to the program detail and make this a non-motion program, meaning disable motion group one.
  - 7 Add the instruction: RO [ x ] = OFF.
  - 8  **NOTE:** [x] is the signal number that opens the gripper.

Completed:

Instructor: \_\_\_\_\_

## Lab 25

### Create a Material Handling Program

---

**Student Name:** \_\_\_\_\_

**Assignment:** The student will:

- Create a program that moves a part
- 

**Condition:** Given a FANUC Robot and a controller

---

- Step:**
- 1 Create a new program called A\_PICK\_PART.
  - 2 Place the cylinder provided by your instructor in the ring marked "PICK POS 1".
  - 3 Starting at HOME.
    - a Jog the robot to a position above the part.
    - b Record that point as a local position.
    - c Name this position "Above Pick".
  - 4 Jog the robot down to the part.
    - a Record that point as PR [6].
    - b Name this PR "Pick Pos".
  - 5 Add an I/O Instruction to close the gripper to pick the part.
  - 6 Return the robot to the "Above Pick" position.
  - 7 Jog the robot to a position above the part drop position.
    - a Record that point as a local position.
    - b Name this position "Above Drop".
  - 8 Jog the robot down to the position to drop the part.
    - a Record that point as PR [7].
    - b Name this PR "Place Pos".
  - 9 Add an I/O Instruction to open the gripper to drop the part.
  - 10 Return the robot to the "Above Drop" position.

See next page for program syntax.

# 14: Inputs and Outputs

- 11 The syntax will be similar to below
  - 1: UFRAME\_NUM=1 ;
  - 2: UTOOL\_NUM=1 ;
  - 3: ;
  - 4:J PR[1:HOME] 100% FINE
  - 5:J P[1: Above Pick] 100% CNT50
  - 6:L PR [6:Pick Pos] 250mm/sec FINE
  - 7: RO[1:Grip Open]=OFF
  - 8:J P[1: Above Pick] 100% CNT50
  - 9:J P[2: Above Place] 100% CNT50
  - 10:L PR [7:Place Pos] 250mm/sec FINE
  - 11: RO[1:Grip Open]=ON
  - 12:J P[2: Above Place] 100% CNT50
  - 13:J PR[1:HOME] 100% FINE
- 12 Test the program.

Completed:

Instructor: \_\_\_\_\_

## 15 BRANCHING

### 15.1 Chapter Objectives

- Branching Instruction Overview
- Program Unconditional Branching instructions
- Program using Data Registers
- Program using Conditional Branching Instructions
- Program using Wait Instructions

### 15.2 Branching Instructions Overview

Branching instructions cause the program to branch, or jump, from one place in a program to another. In other words, a branching instruction controls where the cursor will go next, by controlling the cursor you control how the program will execute.

There are three kinds of branching instructions:

- Label definition instruction
- Unconditional branching instructions
- Conditional branching instructions

### 15.3 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.

```

LBL[x: comment]
 Direct: └─┬─┘ As many as 16 numbers, letters, blank
Label number spaces, the punctuation ;, :, ", (, and),
 and the characters *, _, and @

```

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.

### 15.4 Unconditional Branching

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 (JMP LBL [x]) – Cause the program to branch to a named label.
- Sub program call instructions (CALL) – Cause the program to branch to another program

# 15: Branching

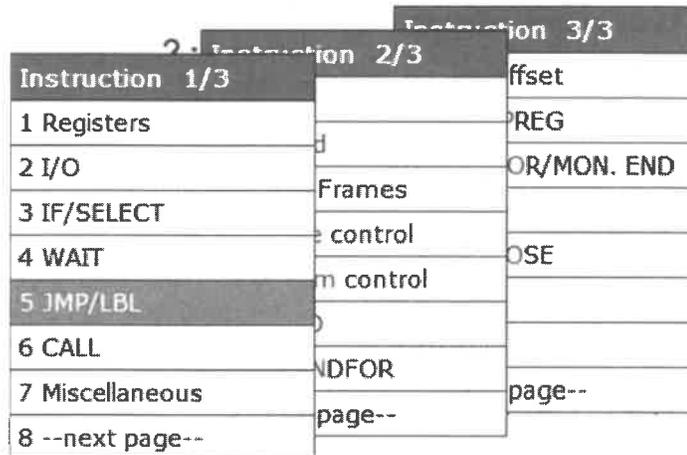
## 15.4.1 Jump Instruction

The JMP LBL[x] instruction causes the program to branch to the specified label. This instruction can cause the program to jump up or down within the current program. Generally, when used as an unconditional branch instruction, the JMP LBL [x] instruction will jump up in the program.

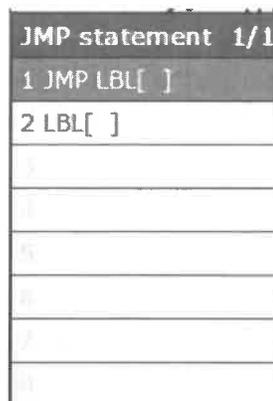
The label instruction defines the destination that the cursor is to “jump” to. The label instruction must be contained within the current program and have the same ID as the Jump Instruction.

### Procedure 15-1 Programming a Jump Instruction

- 1 If necessary, place a blank line in the program at the point to add the Jump Instruction.
- 2 Press F1, [INST]
  - a If F1 does not display [INST], press NEXT
- 3 Cursor down to JMP/LBL, press ENTER.



- 4 Choose JMP LBL [ ]



- 5 Enter the value for the label using the numeric keypad.

| FANUC_ACADEMY                                                                                                                                                                                                                                           | FANUC_ACADEMY                                                                                                                                                                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11/13                                                                                                                                                                                                                                                   | 12/13                                                                                                                                                                                                                              |
| <pre> 2: UTOOL_NUM=1 3: 4:J PR[1:HOME] 100% FINE 5: 6:J P[1:Square] 100% CNT50 7:L P[2] 6000cm/min CNT50 8:L P[3] 750mm/sec CNT50 9:L P[2] 750mm/sec CNT50 10:L P[1:Square] 750mm/sec CNT50 11: JMP LBL[ ] 12:J PR[1:HOME] 100% FINE Enter value </pre> | <pre> 3: 4:J PR[1:HOME] 100% FINE 5: 6:J P[1:Square] 100% CNT50 7:L P[2] 6000cm/min CNT50 8:L P[3] 750mm/sec CNT50 9:L P[2] 750mm/sec CNT50 10:L P[1:Square] 750mm/sec CNT50 11: JMP LBL[1] 12:J PR[1:HOME] 100% FINE [End] </pre> |

- 6 Add the corresponding Label (LBL [ ]) Instruction at the destination.
- 7 Follow the same procedure as adding the JMP LBL, but choose LBL.
- a Define the LBL value as the same as the JMP LBL value.

```

FANUC_ACADEMY
6/13
3:
4:J PR[1:HOME] 100% FINE
5: LBL[1]
6:J P[1:Square] 100% CNT50
7:L P[2] 6000cm/min CNT50
8:L P[3] 750mm/sec CNT50
9:L P[2] 750mm/sec CNT50
10:L P[1:Square] 750mm/sec CNT50
11: JMP LBL[1]
12:J PR[1:HOME] 100% FINE
[End]

```

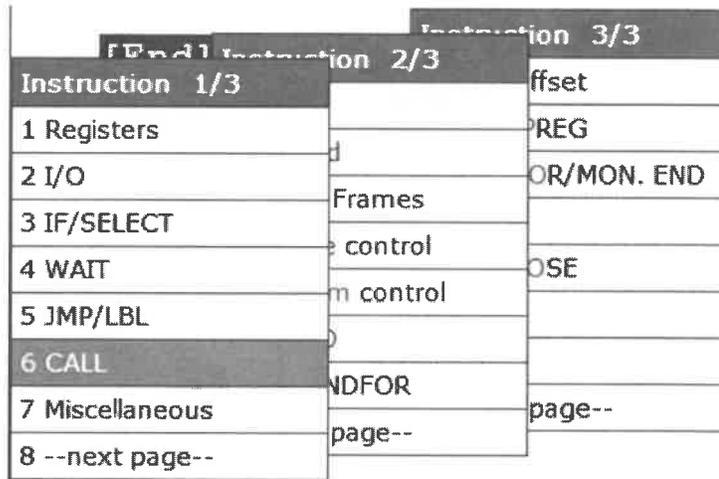
### 15.4.2 CALL Instruction

The CALL Instruction, causes the program to branch to a remote program or sub-routine. The program is executed then the motion control returns to the main program on the line immediately following the CALL instruction. It is **not** necessary to add a call statement in the second program to return back to the first program as it will automatically return when it reaches the program END.

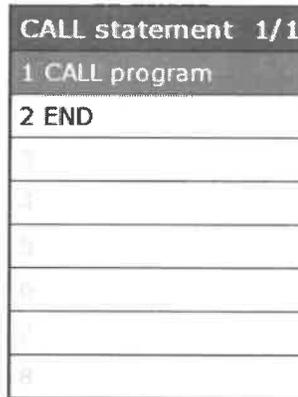
#### Procedure 15-2 Programming a Call Instruction

- 1 If necessary, place a blank line in the program at the point to add the CALL Instruction.
- 2 Press F1, [INST]
  - a If F1 does not display [INST], press NEXT
- 3 Cursor down to CALL, press ENTER.

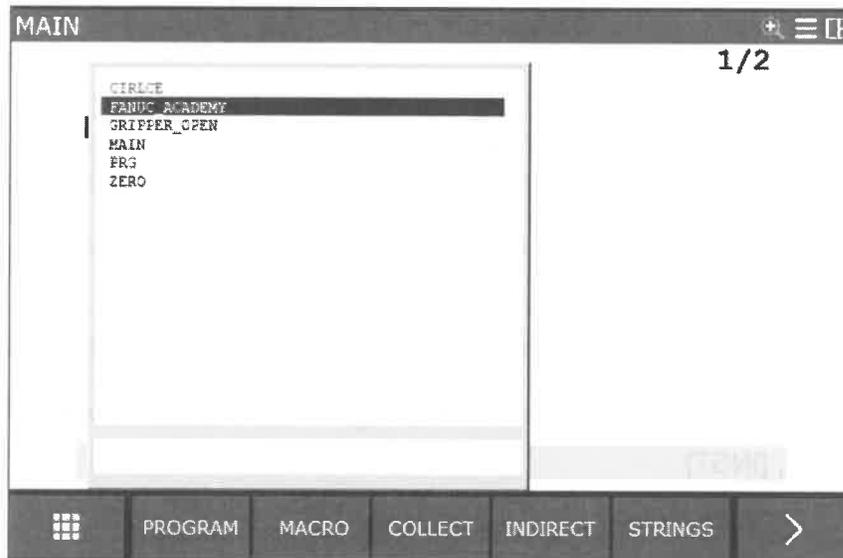
# 15: Branching



4 Choose CALL program.



5 Select the program to call.



- 6 The instruction will be added to the program.

```

MAIN
1: CALL FANUC_ACADEMY
[End]
1/2

```

## 15.5 Data Registers

A Data Register is a storage space for a number. What that number is used for is completely dependent on the programmer. The Data Register can be used to track part counts, speed values cycle counts, timers, etc. The number can be a real number or an integer.

Up to 999 Data Registers are available for all the programs in the controller combined, the default number is 200. Much like Position Registers, Data Registers are globally accessible and identified by R [x], where “x” is the number of the Data Register.

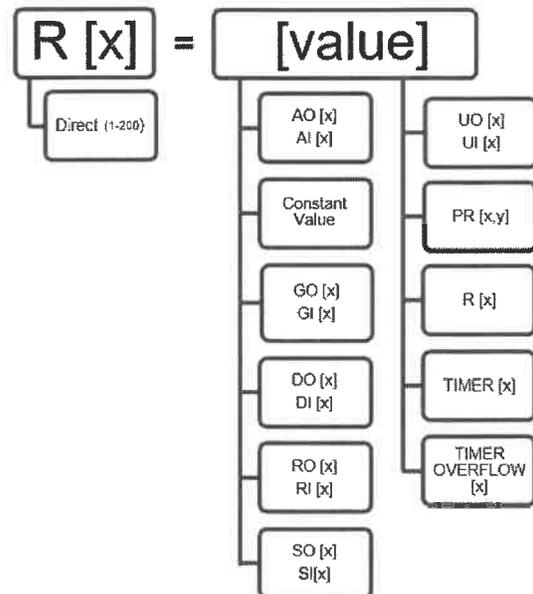
Many instructions employ direct or indirect addressing techniques. When direct addressing is used, the actual value is entered into the instruction. For example, if the Data Register instruction R[3]= 0 is used, the current contents of Data Register 3 is replaced with the value of 0.

You can also perform mathematical operations using Data Register instructions. These instructions are useful to manipulate a value that has been entered into a Data Register. For example, if the Data Register instruction R [3] = R [3] + 1 is used, the current contents of the Data Register R [3] would be increased by one. Each time the instruction is executed the contents would increase.

### 15.5.1 Direct Addressing Data Register Instructions

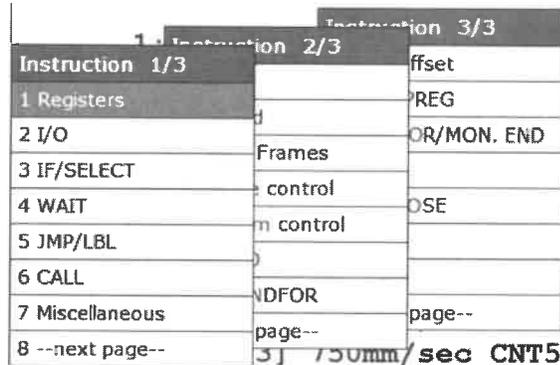
Many instructions employ direct or indirect addressing techniques. When direct addressing is used, the actual value is entered into the Data Register instruction. For example, if the Data Register instruction R [3] = 2 is used, the current contents of Data Register 3 is replaced with the value 2.

**NOTE:** Indirect addressing using data registers will not be covered in this training manual.

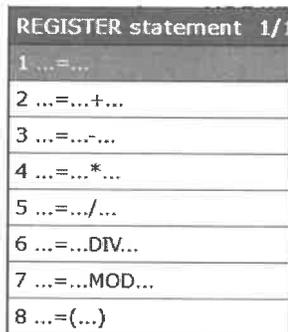


## Procedure 15-3 Programming a Data Register Instruction

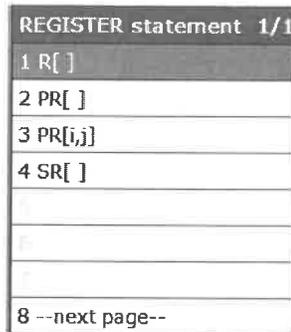
- 1 If necessary, place a blank line in the program at the point to add the Data Register Instruction.
- 2 Press F1, [INST].
  - a If F1 does not display [INST], press NEXT
- 3 Select Register



- 4 Choose the first template ( ... = ... )
  - a The software will prompt you to fill in the necessary items.



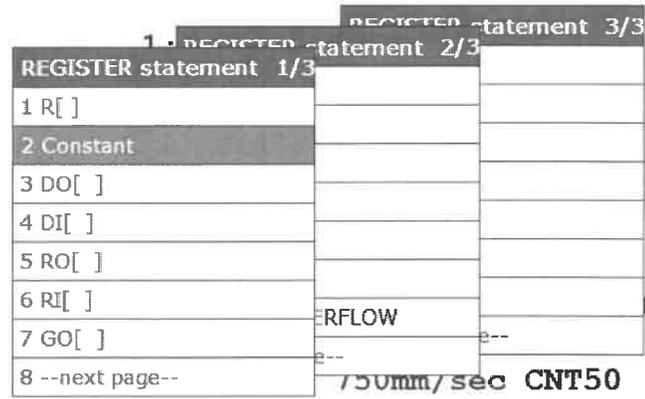
- 5 Chose Data Register (R [ ]) as the register type.



- 6 Input the value of the register to use.

4: R[ ] = ...

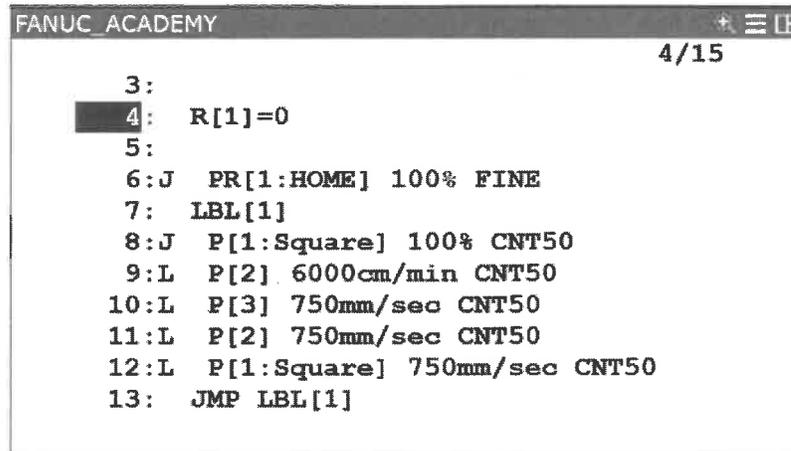
- 7 Choose the variable type that the register will reflect.
  - a If setting the register to reflect a number select CONSTANT.



- 8 Enter the value that the register will reflect.

4: R[1]=Constant → 4: R[1]=0

- 9 When the program executes R [1] will be set to reflect a zero.



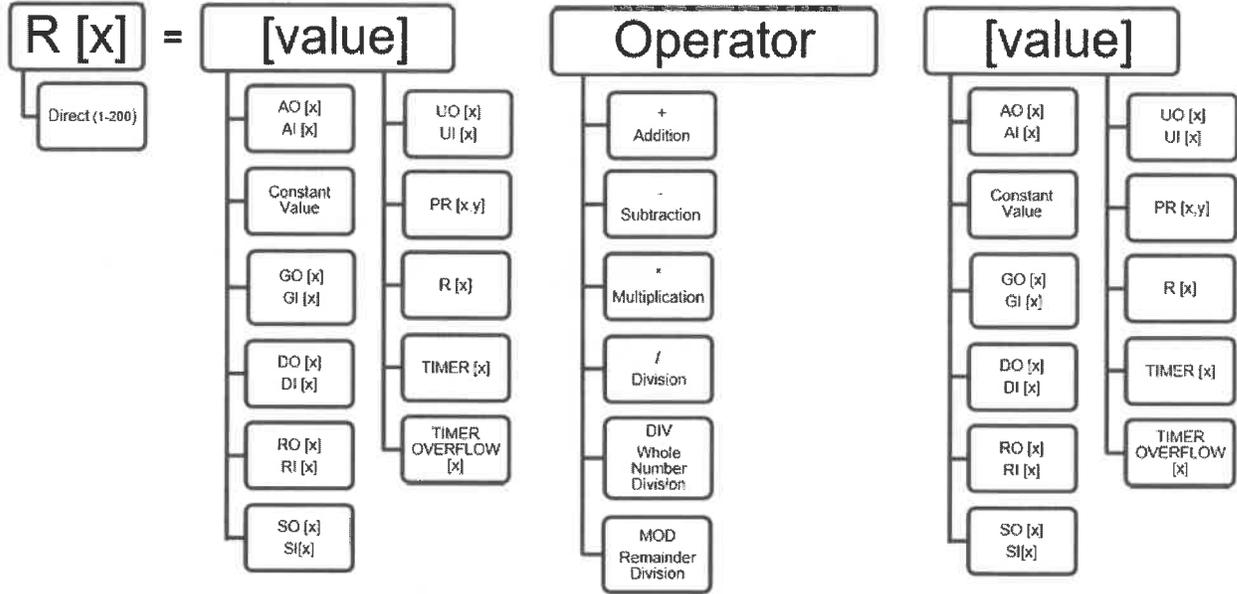
### 15.5.2 Performing Arithmetic Operations with Register Instructions

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)

# 15: Branching

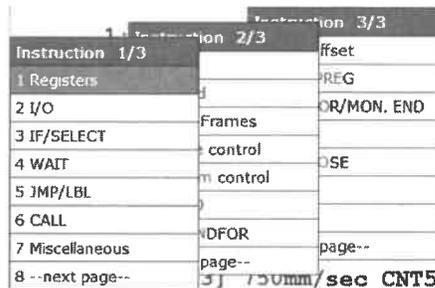


You can use multiple arithmetic operators in a single instruction. However, there are the following limitations:

- You can mix + and – in the same instruction. Arithmetic operations within an instruction that mixes + and – will be performed from left to right. You cannot mix \* or / in an instruction that already contains + or –.
- You can mix \* and / in the same instruction. Arithmetic operations within an instruction that mixes \* and / will be performed from left to right. You cannot mix + or – in an instruction that already contains \* or /.
- The maximum number of arithmetic operators you can have in the same instruction is 5.

### Procedure 15-4 Program Data Register Instructions with Arithmetic Operations

- 1 If necessary, place a blank line in the program at the point to add the Data Register Instruction.
- 2 Press F1, [INST].
  - a If F1 does not display [INST], press NEXT
- 3 Select Register



- 4 Choose the desired arithmetic operation template.
  - a The software will prompt you to fill in the necessary items.
  - b If you choose the incorrect arithmetic operator it can be changed at any time by highlighting the operator and pressing F4, [CHOICE].

| REGISTER statement 1/1 |               |
|------------------------|---------------|
| 1                      | ...=...       |
| 2                      | ...=...+...   |
| 3                      | ...=...-...   |
| 4                      | ...=...*...   |
| 5                      | ...=.../...   |
| 6                      | ...=...DIV... |
| 7                      | ...=...MOD... |
| 8                      | ...=(...)     |

- 5 Chose Data Register (R [ ]) as the register type.

| REGISTER statement 1/1 |               |
|------------------------|---------------|
| 1                      | R[ ]          |
| 2                      | PR[ ]         |
| 3                      | PR[i,j]       |
| 4                      | SR[ ]         |
| 5                      |               |
| 6                      |               |
| 7                      |               |
| 8                      | --next page-- |

- 6 Input the value of the data register to use.

**13: R[ ] = ... + ...**

- 7 Choose the variable type that the field will utilize.
  - a If using the instruction as a counter select the same register.

| REGISTER statement 1/3 |               | REGISTER statement 2/3 |  | REGISTER statement 3/3 |  |
|------------------------|---------------|------------------------|--|------------------------|--|
| 1                      | R[ ]          |                        |  |                        |  |
| 2                      | Constant      |                        |  |                        |  |
| 3                      | DO[ ]         |                        |  |                        |  |
| 4                      | DI[ ]         |                        |  |                        |  |
| 5                      | RO[ ]         |                        |  |                        |  |
| 6                      | RI[ ]         |                        |  |                        |  |
| 7                      | GO[ ]         |                        |  |                        |  |
| 8                      | --next page-- |                        |  |                        |  |

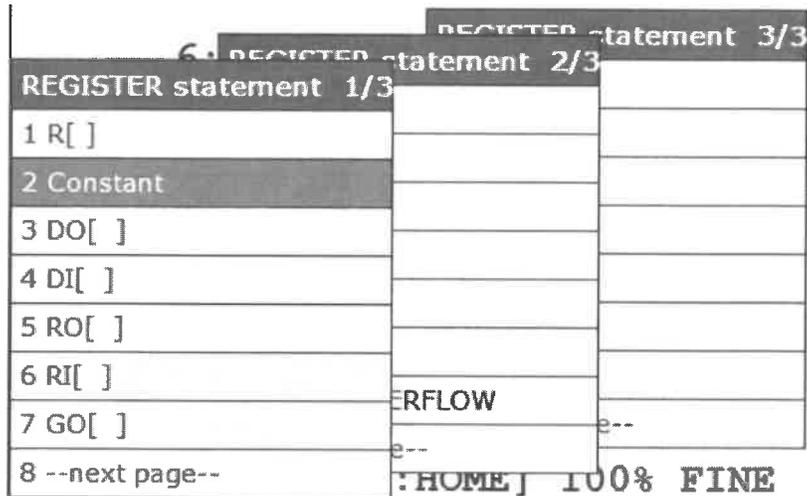
# 15: Branching

8 Enter the value that the register will reflect.

**13: R[1]=R[1] + ...**

9 Select the variable to increment the instruction.

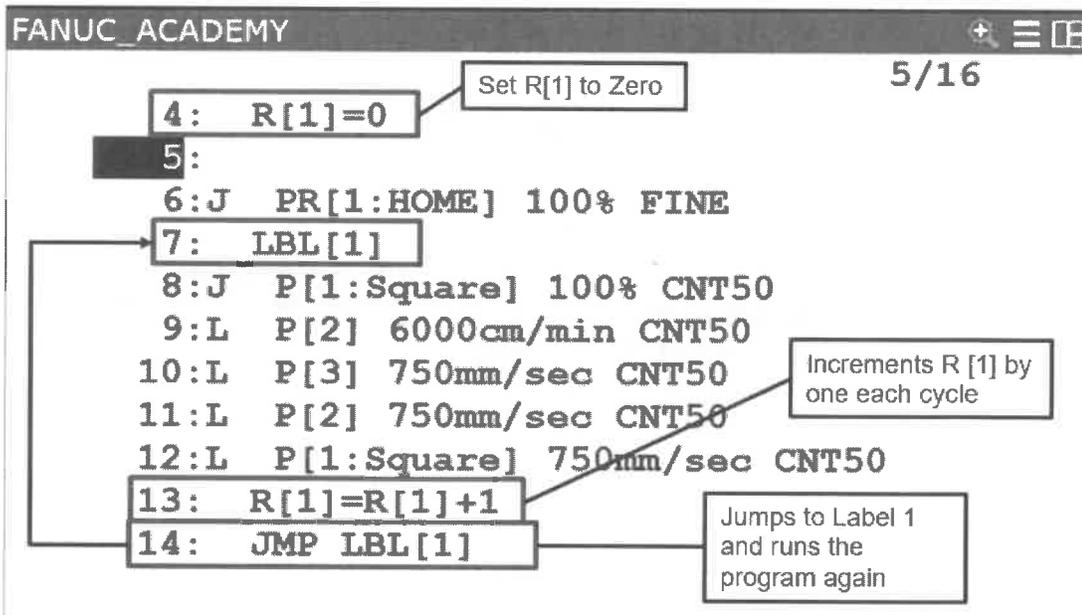
a If incrementing by a set value, choose CONSTANT.



10 Enter the value for the selected variable.

**13: R[1]=R[1]+Constant** → **13: R[1]=R[1]+1**

11 When the program executes the number of cycles will be counted.



## 15.6 Conditional Branching

Conditional branching instructions branch from one place to another within a program or send program flow to a subroutine, 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.
  - Can base conditions on a Data Register, I/O, or System Variables.
- SELECT instructions – Branch to one of several jump or call instructions, depending on the value of a register.
  - You can only look to the contents of a register with a SELECT instruction.

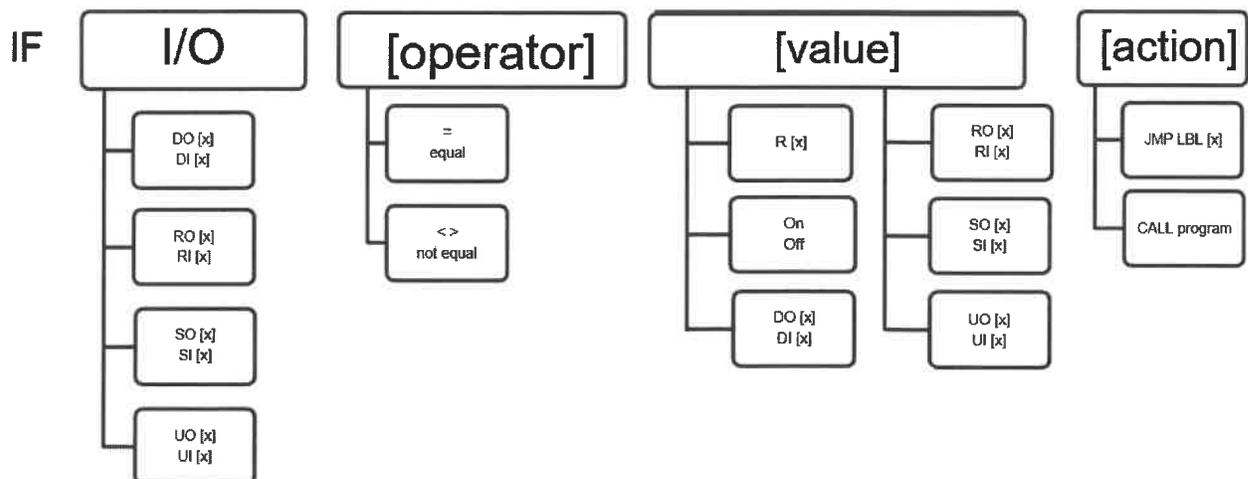
### 15.6.1 IF Instructions

An IF Instruction compares the value contained in a register, an input, an output, or system variable with another value and take an action if the comparison is true.

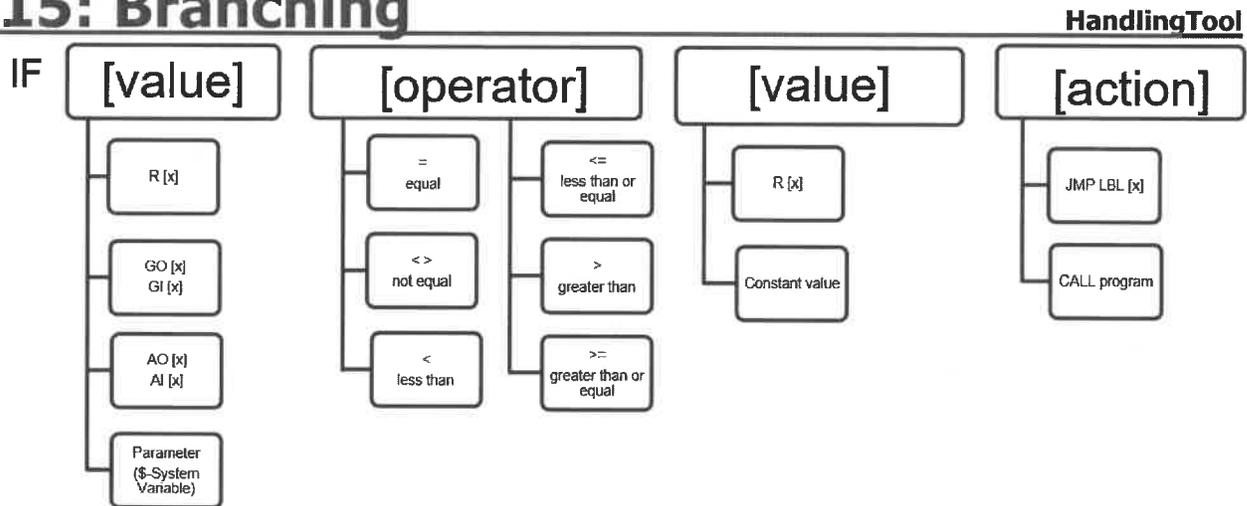
- There are only two possible actions when the IF Instruction is true.
  - CALL program
  - JMP LBL.
- When the condition specified in the IF Instruction is satisfied, the action is executed.
- When the condition specified in the IF Instruction is **not** satisfied, the action is not executed and the program continues on the next line.

### 15.6.2 Types of IF Instructions

- The variables available when programming an IF Instruction are dependent on the specified condition.
  - The software will only allow the programmer to select items that agree with the specified condition.



# 15: Branching



 **NOTE:** Items available for the IF instruction are dependent on the first variable item selected.

## 15.6.3 IF Instruction Limitations

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 operator
  - IF [cond1] OR [cond2] OR ..., [action]
  - For example, 1: IF DI[10]=ON OR R[7]=R[8], JMP LBL[2]

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.

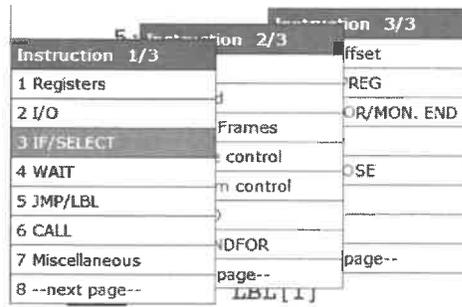
- TPIF-062 AND operator was replaced to OR
- TPIF-063 OR operator was replaced to AND

The maximum number of logical conditions that can be taught in the same operation is 5.

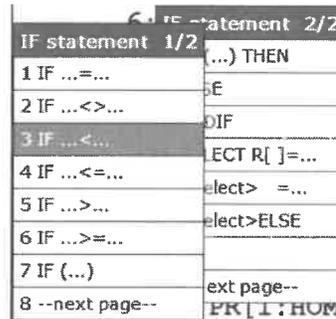
## 15.6.4 Programming an IF Instruction

### Procedure 15-5 Programming an IF Instruction

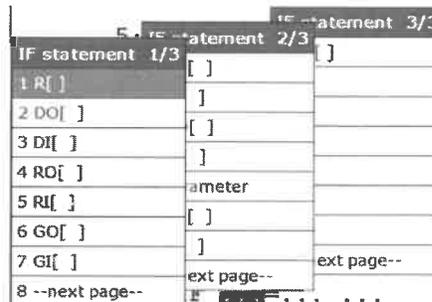
- 1 Place the cursor at the location in the program to add the IF Instruction.
- 2 Press F1, [INST].
  - a If F1 does not display [INST], press NEXT
- 3 Select IF/SELECT



- 4 Choose the desired operation template.
  - a The software will prompt you to fill in the necessary items.
  - b If you choose the incorrect operator template it can be changed at any time by highlighting the operator and pressing F4, [CHOICE].
  - c The condition for this example is if the robot has completed less than the desired number of cycles.



- 5 Chose the variable type to set the condition.
  - a This IF Instruction condition will be based on number of cycles completed. The count is stored in R [1].

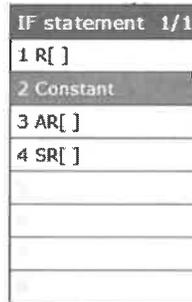


- 6 Input the value of the data register to use.

**14: IF R[ ]=... ..**

- 7 Choose the variable that the condition specified above will compare.
  - a For this example set the number of cycles the robot should complete.

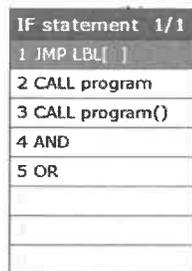
# 15: Branching



8 Enter the value required.

14: IF R[1]<Constant . . . → 14: IF R[1]<5 . . .

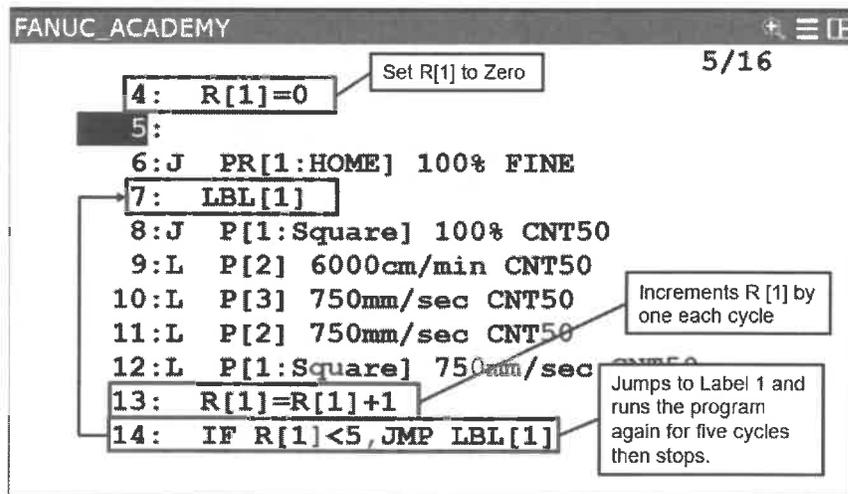
9 Select the action.



10 Specify the JMP LBL number or the program to call.

14: IF R[1]<5, JMP LBL[1]

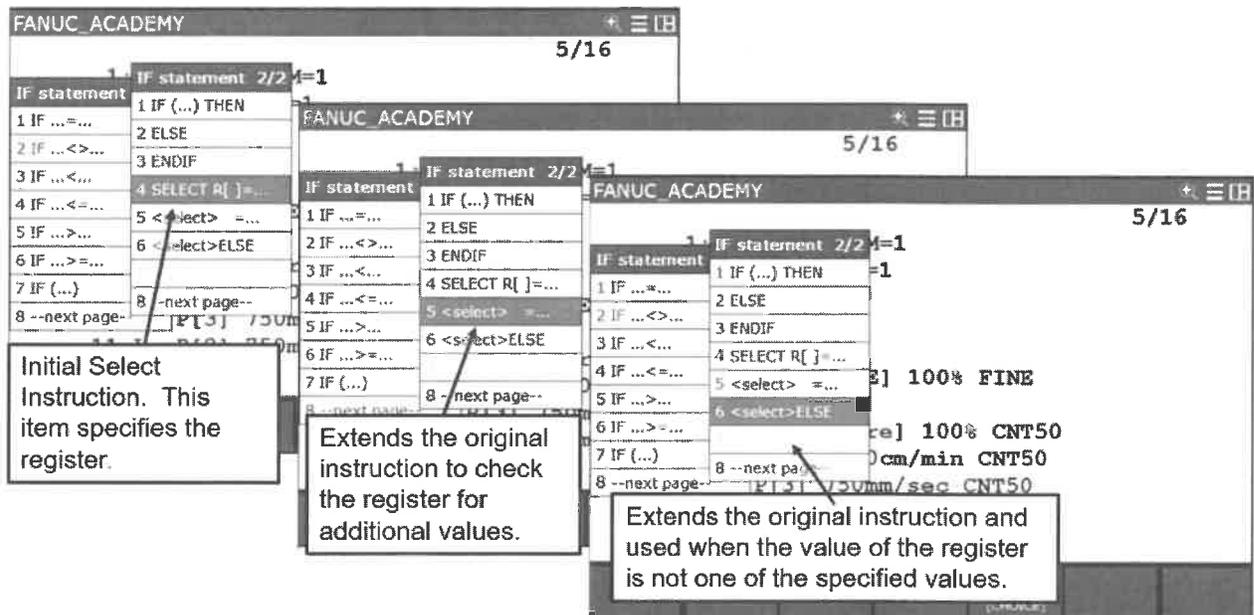
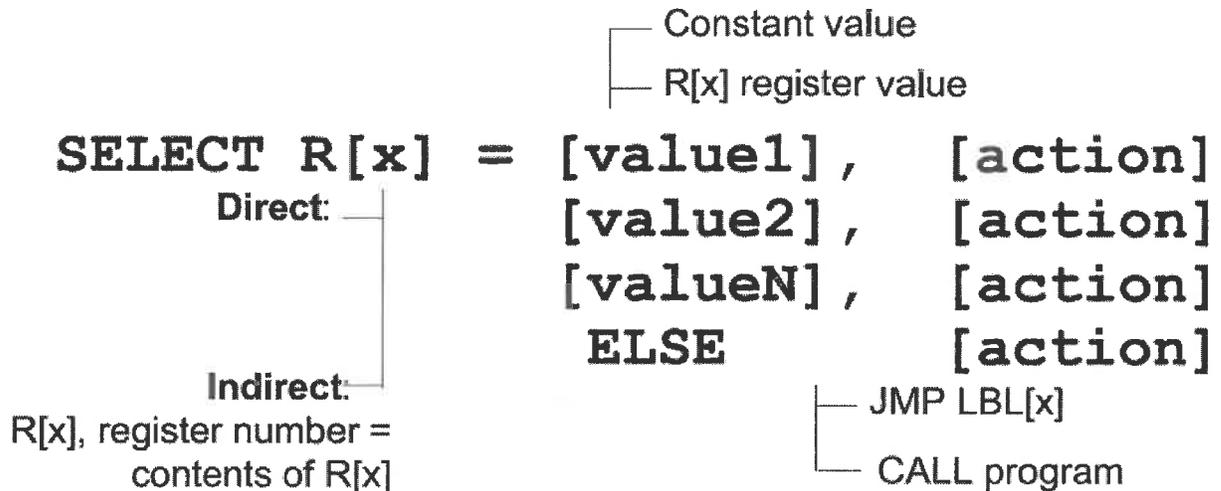
11 The program will execute five cycles and then end.



## 15.6.5 SELECT Instructions

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.

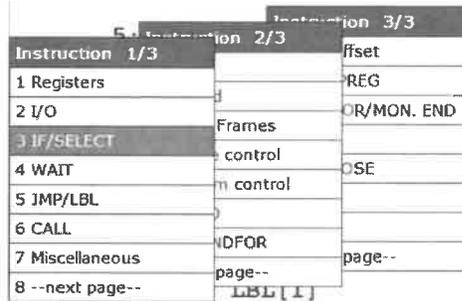


# 15: Branching

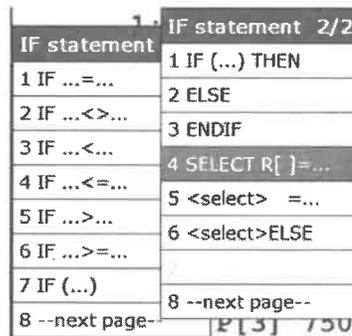
## 15.6.6 Programming a SELECT Instruction

### Procedure 15-6 Programming a SELECT Instruction

- 1 Place the cursor at the location in the program to add the SELECT Instruction.
- 2 Press F1, [INST].
  - a If F1 does not display [INST], press NEXT
- 3 Select IF/SELECT



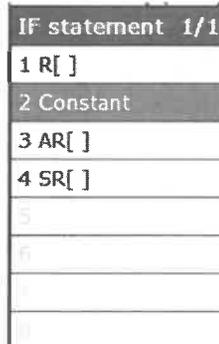
- 4 Choose SELECT R [ ] = ...



- 5 Specify the Data Register.

**2: SELECT R[...]=... ..**

- 6 Select the variable to check the register for.



- 7 Quantify the value.

2: **SELECT R[1]=Constant** ... → 2: **SELECT R[1]=1** ...

8 Choose the action.

| IF statement 1/1 |                |
|------------------|----------------|
| 1                | JMP LBL[ ]     |
| 2                | CALL program   |
| 3                | CALL program() |
|                  |                |
|                  |                |
|                  |                |
|                  |                |
|                  |                |

9 Specify the JMP LBL destination or the program to CALL.

```

MAIN
3/3
1: LBL[1]
2: SELECT R[1]=1,CALL SQUARE
[End]

```

10 To check the Data Register for another value and specify another condition Press F1, [INST]

- a Choose IF/SELECT
- b Choose <select>

**NOTE:** This instruction means check the Data Register specified above.

| IF statement 1/1 |  | IF statement 2/2  |  |
|------------------|--|-------------------|--|
| IF statement     |  | 1 IF (...) THEN   |  |
| 1 IF ...=...     |  | 2 ELSE            |  |
| 2 IF ...<>...    |  | 3 ENDIF           |  |
| 3 IF ...<...     |  | 4 SELECT R[ ]=... |  |
| 4 IF ...<=...    |  | 5 <select> =...   |  |
| 5 IF ...>...     |  | 6 <select>ELSE    |  |
| 6 IF ...>=...    |  |                   |  |
| 7 IF (...)       |  | 8 --next page--   |  |
| 8 --next page--  |  |                   |  |

11 Specify the next value to check the Data Register for.

```

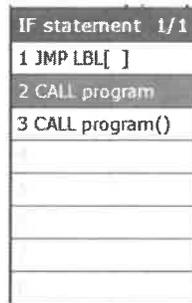
IF statement 1/1
1 R[]
2 Constant
3 AR[]
4 SR[]

```

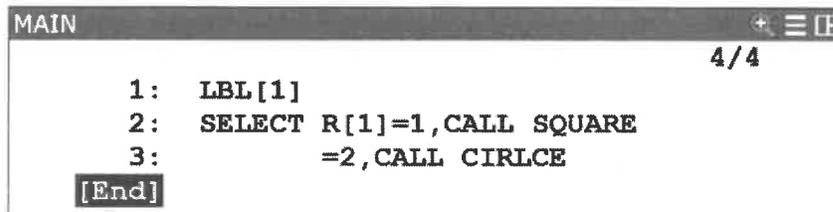
LBL[1]  
**SELECT R[1]=1,CALL SQUARE**  
 =... ..

# 15: Branching

- 12 Quantify the value.
- 13 Choose the action.

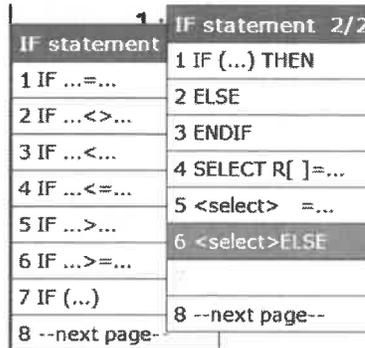


- 14 Specify the JMP LBL destination or the program to CALL.

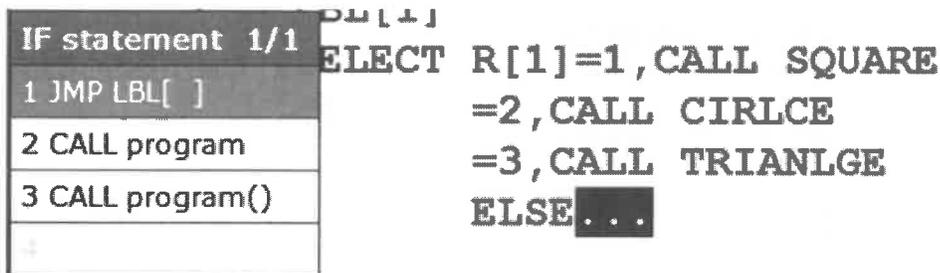


- 15 Repeat STEPS 10-14 to continue specifying more conditions and actions.
- 16 To check the Data Register for any values not specified Press F1, [INST]
  - a Choose IF/SELECT
  - b Choose <select> ELSE

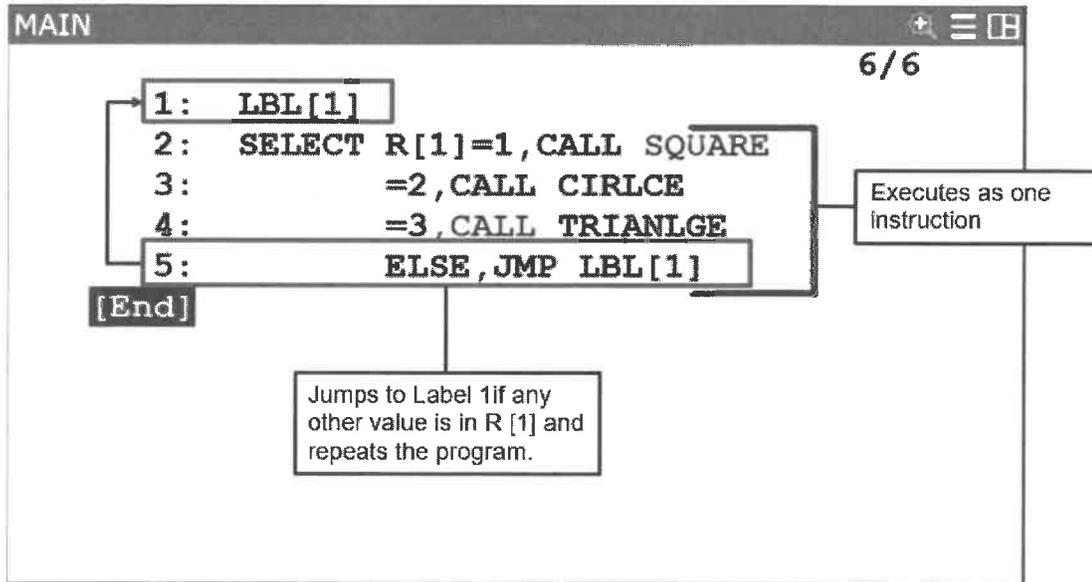
**NOTE:** This instruction means check the Data Register for every value not specified.



- 17 Choose the action.



- 18 Specify the JMP LBL destination or the program to CALL.
- 19 The program will execute a called program based on the value of the Data Registered specified.
  - a If none of the conditions are true the program will continue to loop.
  - b If one of the programs are executed it will be ran one time and then the program will end.



## 15.7 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.

### 15.7.1 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.

#### 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]

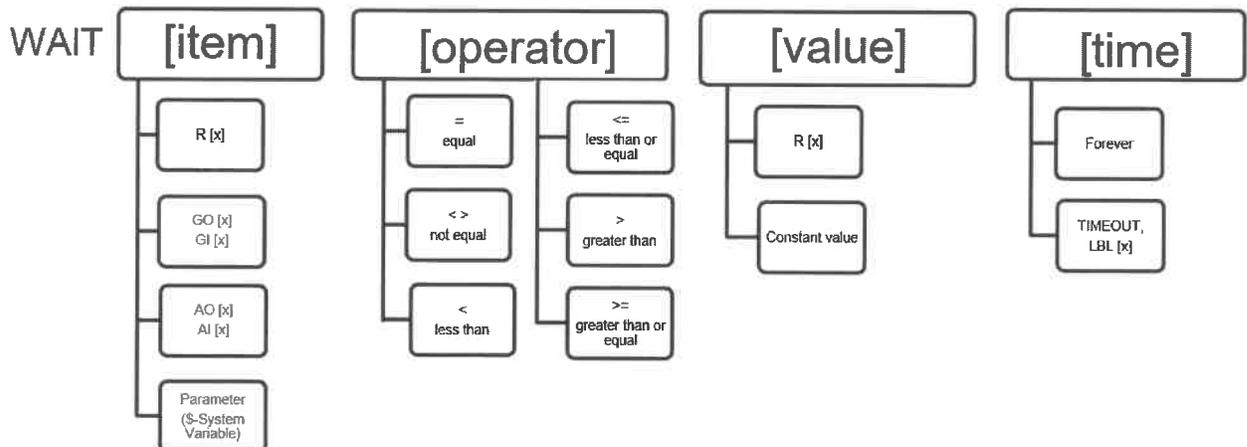
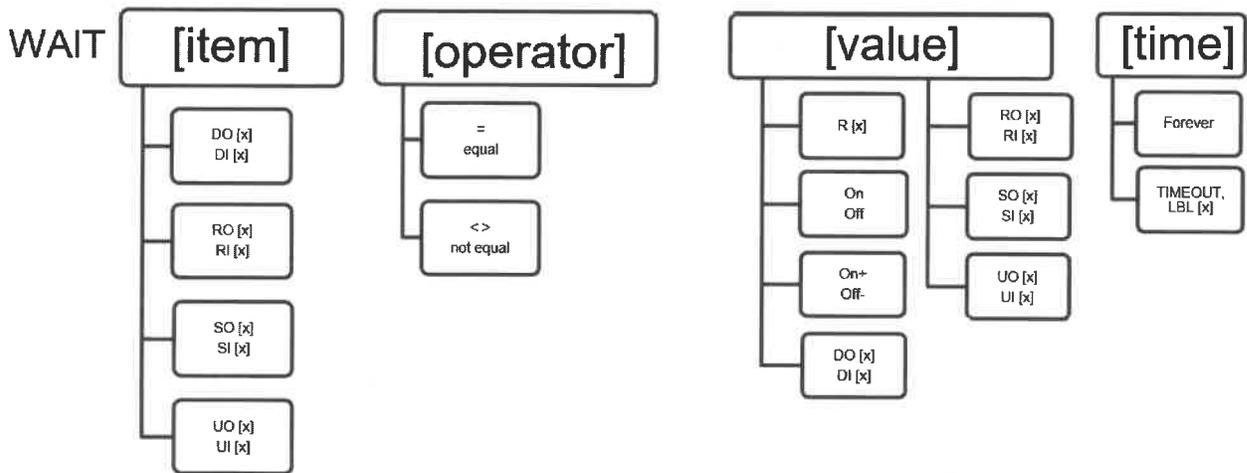
## 15.7.2 WAIT Conditions

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, this is specified by default
- 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.

**NOTE:** WAIT with TIMEOUT, LBL [ ] will not be covered further in this manual.

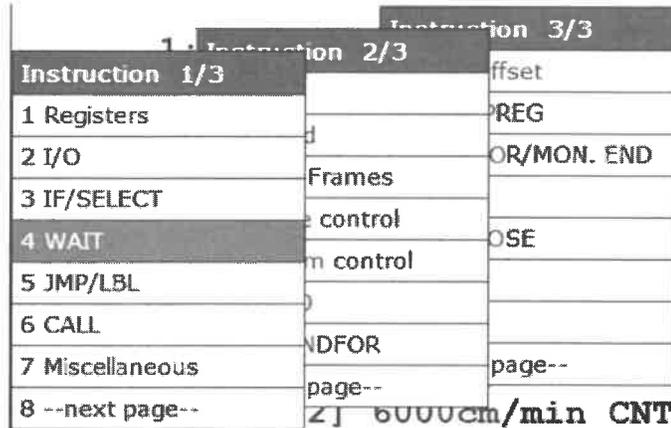
- WAIT Instructions can contain multiple conditions using “AND” or “OR” operators.
  - The limitations and rules are the same as an IF Instruction.
- The items available to specify when programming a WAIT Instruction are dependent on what sets the condition.



## 15.7.3 Programming a WAIT Instruction

### Procedure 15-7 Programming a WAIT Instruction

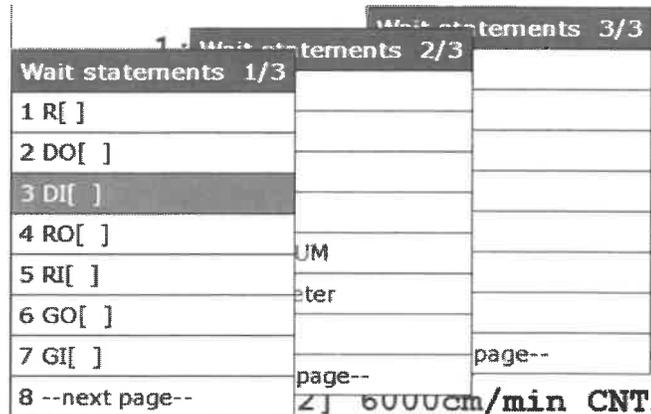
- 1 Place the cursor at the location in the program to add the WAIT Instruction.
- 2 Press F1, [INST].
  - a If F1 does not display [INST], press NEXT
- 3 Select WAIT



- 4 Choose the type of WAIT Instruction.



- 5 Specify the condition for the WAIT Instruction.

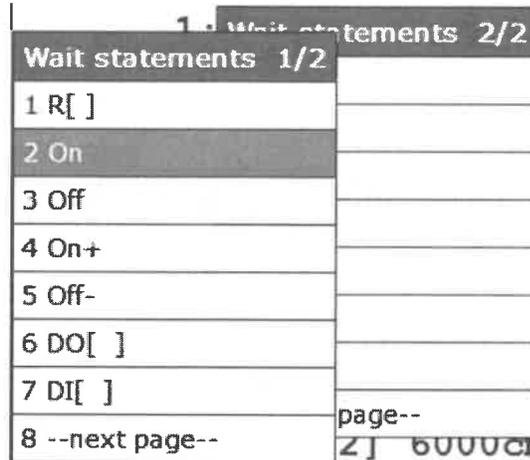


# 15: Branching

6 Quantify the selected variable.

**8: WAIT DI [ ] = ...**

7 Select the condition to be satisfied.

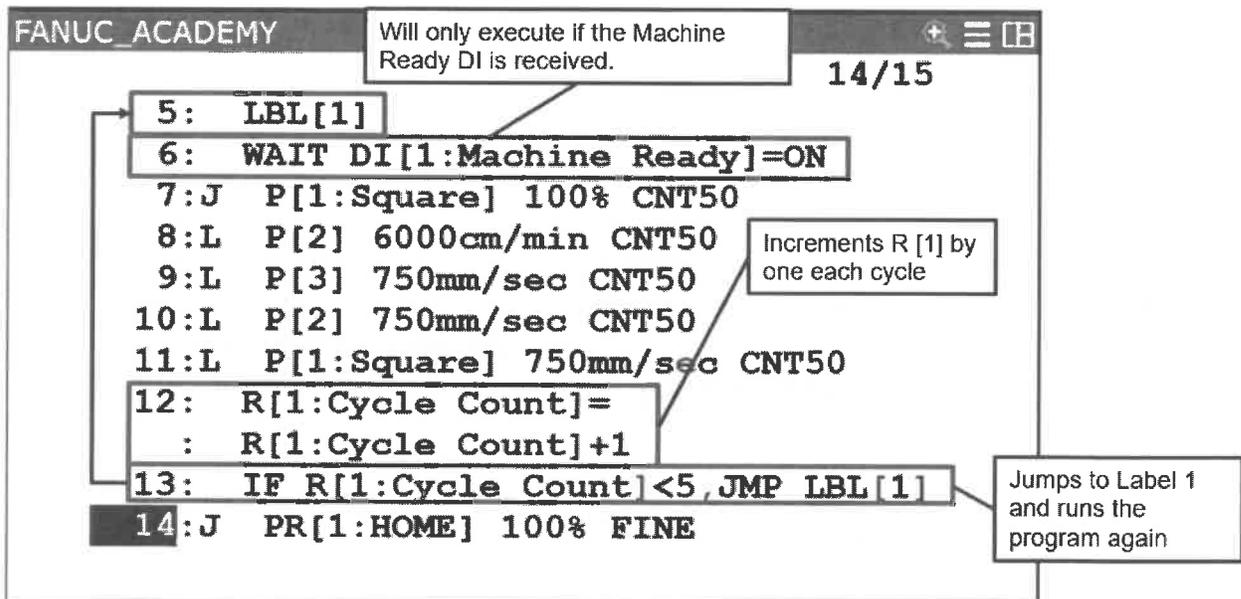


8 The instruction will be added to the program.

**8: WAIT DI [1]=ON**

9 The program will wait for the condition specified in the WAIT instruction is satisfied.

a For this example the program will wait at line 6 until the DI is turned on.



**15.8 Chapter Review**

- 1. What instruction will send program flow to a sub-program**
  - A JMP LBL
  - B CALL
  - C IF
  - D SELECT
  
- 2. What are the two possible actions of an IF/SELECT statement?**
  - A Select & Set an output
  - B Set an Output & CALL
  - C JMP LBL & CALL
  - D JMP LBL & WAIT
  
- 3. Which instruction would increment Data Register 10 (R [10] by two (2)?**
  - A R [10] = 2
  - B R [10] ++
  - C R [10] = R[10] +2
  - D R [10] +2

## Lab 26

### Program with Branching

---

**Student Name:** \_\_\_\_\_

**Assignment:** The student will:

- Create a program that calls a group of programs in a specified sequence utilizing conditional branching instructions.
- 

**Condition:** Given a FANUC Robot and a controller

---

- Step 1** Create a new program called MAIN1. This program will:
- a Loop forever until a specified input has been received. (See Section 15.6.4 Programming an IF Instruction, pg. 258; Section 15.7.3 Programming a WAIT Instruction, pg. 267; and Section 15.4.1 Jump Instruction, pg.248)

**MAIN1**

```
1: LBL[1]
2: IF DI[2]=ON, CALL A_SQUARE
3: WAIT 1.00(sec)
4: IF DI[3]=ON, CALL A_CIRCLE
5: WAIT 1.00(sec)
6: IF DI[4]=ON, CALL A_TRIANGLE
7: JMP LBL[1]
[END]
```

- b Once one of the specified inputs is turned on, the associated program will execute until the input is turned off or another specified condition is true.
- 2** Verify the program executes as expected.

Completed:

Instructor: \_\_\_\_\_

## Lab 27

### Programming with Branching

---

**Student Name:** \_\_\_\_\_

**Assignment:** The student will:

- Create a program that calls a group of programs in a specified sequence utilizing conditional branching instructions.
- 

**Condition:** Given a FANUC Robot and a controller

---

- Step:**
- 1 Create a Group Input using DI [5-7]
  - 2 Create a new program called MAIN2. This program will:
    - a Loop forever until a value within a specific range has been entered on the program select register. (See Section 15.6.6 Programming a SELECT Instruction, pg. 262)

**MAIN2**

```
1: LBL[1]
2: R[5]=GI[1]
3: SELECT R[5]=1,CALL A_SQUARE
4: =2,CALL A_CIRCLE
5: =3,CALL A_TRIANGLE
6: ELSE, JMP LBL[1]
```

[END]

- 3 Once the program has captured a valid number, it will execute this program once and then it will move on to the next instruction
- 4 Verify the program executes as expected.

Completed:

Instructor: \_\_\_\_\_

## Lab 28 Register Instructions

---

Student Name: \_\_\_\_\_

**Assignment:** The student will:

- Create a program that runs a specific number of cycles and then stops utilizing conditional branching instructions.
- 

**Condition:** Given a FANUC Robot and a controller

---

- Step:**
- 1 Copy A\_SQUARE and name the new program A\_COUNTER.
  - 2 Modify this program to cycle a predefined number of cycles and then end program.
  - 3 To accomplish this you need to:
    - a Assign a Data Register at the DATA screen as your counter register. (See Section 15.5.1 Direct Addressing Data Register Instructions, pg. 251)
    - b Initialize the Data Register outside the main loop within the program.
    - c Increment Data Register within each loop. (See Section 15.5.2 Performing Arithmetic Operations with Register Instructions, pg. 253)
    - d Use a conditional branching instruction to monitor the number of cycles executed while the program runs. (See Section 15.6.4 Programming an IF Instruction, pg. 258)

### A\_COUNTER

```
1: UFRAME_NUM=1
2: UTOOL_NUM=1
3:
4: R[1]=0
5:
6:J PR[1] 100% FINE ;
7: LBL[1]
8:
9:J P[1:Square] 100% CNT50
```

(PROGRAM CONTINUES ON NEXT PAGE)

```
10:L P[2] 6000cm/min CNT50
11:L P[3] 750mm/sec CNT50
12:L P[4] 750mm/sec CNT50
13:L P[1:Square] 750mm/sec CNT50
14:
15: R[1]=R[1]+1
16: IF R[1]<5,JMP LBL[1]
17:
18:J PR[1] 100% FINE
```

- 4 Verify program runs as expected.

Completed:

Instructor: \_\_\_\_\_



## 16 MACROS

### 16.1 Chapter Review

- Create a Macro Program
- Assign a Macro
- Use Macro Instruction

### 16.2 Macro Overview

A macro is typically an easily initiated, concise program used to perform an often repeated function.

Macro programs can be executed from:

- Specific teach pendant keys:
- Specific Standard Operator Panel buttons:
  - SOP 4 and SOP 5

 **NOTE:** SOP macro assignments can only be made on RJ3iB and **older** controllers

- The MANUAL FUNCTIONS macro screen
- Issuing an input signal:
  - DI, RI, and UI
- Within a program using the MACRO COMMAND instruction

To use a macro, you must

- Write the macro program
- Set up the macro to define how it will be executed
- Execute the macro

 **NOTE:** System level macros are application-specific instructions that are predefined and cannot be changed by the user. These macros are identified with the letter “s” at the far right side of the Macro SETUP screen. You cannot change instruction name or program for system level macros on any screen.

### 16.3 Creating a Macro

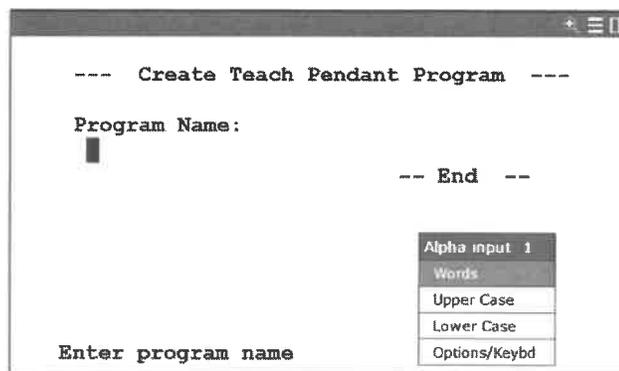
You can create your own Macros for repetitive tasks or any purpose you want. Any program can be assigned as a Macro. Therefore, creating a Macro is very similar to creating a regular Teach Pendant program.

If you press F2, DETAIL from the Select menu, you will be able to modify the program's subtype to designate as a Macro.

- If you create a program from the Select menu with F1,[TYPE] set to Macro, the created program will automatically be given the subtype Macro
- When setting up the Macro Table, any program you place into the Table will automatically be given the subtype Macro.

### Procedure 16-1 Creating a Macro Program

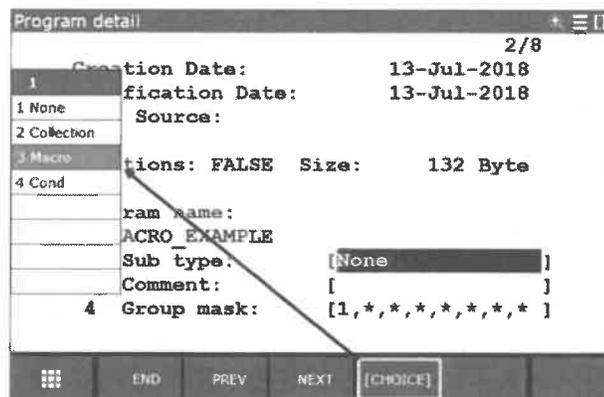
- 1 Press the Select key.
- 2 Press F2, CREATE. If F2 is not displayed press the next key.



- 3 Enter the name of the program, the program name should be something descriptive of what the program is to do.

**NOTE:** Naming conventions are the same as previously discussed.

- 4 Once named, press F2, DETAIL.
- 5 Cursor to the sub type line and press F4, [CHOICE]
- 6 Select MACRO from the list.



- 7 With the assignment set to Macro, press F2, END
  - a Add instructions to the newly created Macro program as necessary.

## 16.4 Macro Assignments

Before a Macro can be a “fully functioning” Macro, it must be assigned on the Macro table. There are a number of different assignments that can be allocated to a Macro program. They are generally broken up between Manual Mode Assignments and Automatic Mode Assignments.

- Manual Mode Assignments:
  - Teach Pendant User Keys
    - User Key 1-7
    - Shift User Key 1-7
  - Manual Functions
- Automatic Mode Assignments
  - Issuing an input signal:
    - DI, RI, and UI
  - Specific Standard Operator Panel buttons:
    - SOP 4 and SOP 5

 **NOTE:** SOP macro assignments can only be made on RJ3iB and **older** controllers

### 16.4.1 Teach Pendant User Key Assignments

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.**

- You can define up to seven macro commands to run when the user key is pressed alone (UK [1] – UK [7]).
  - You can define up to 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.
- 

UK [1] and  
SU [1]



UK [2] and  
SU [2]



UK [3] and  
SU [3]



UK [4] and  
SU [4]
- 

UK [7]  
and  
SU [7]



UK [6]  
and  
SU [6]

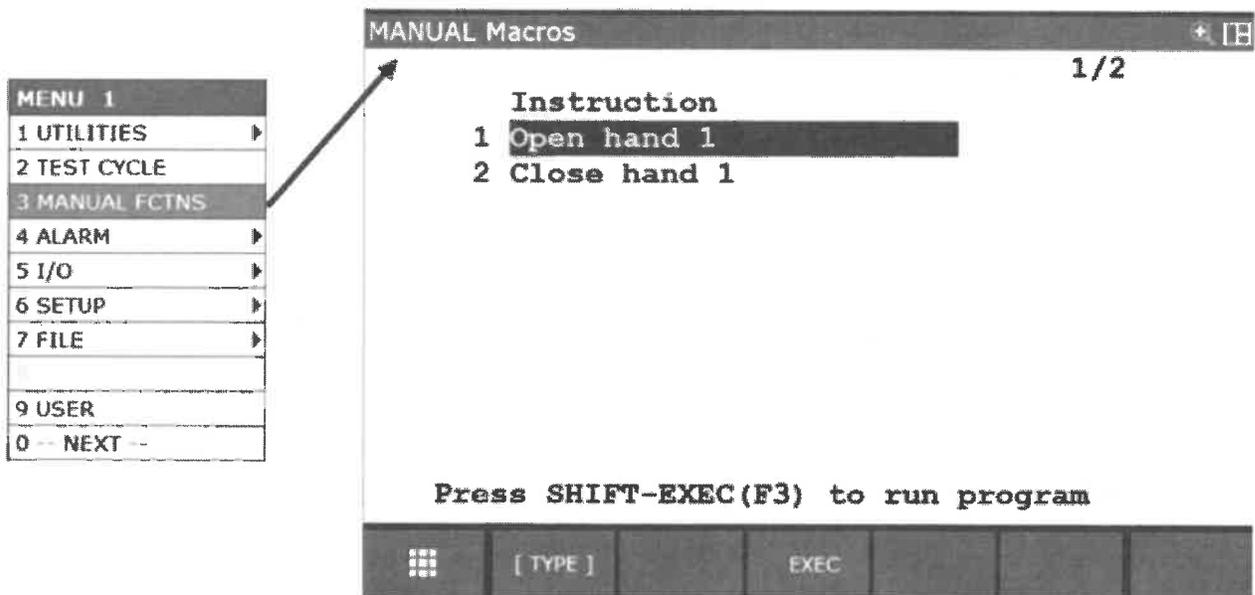


UK [5]  
and  
SU [5]

## 16.4.2 MANUAL FUNCTIONS Assignments

You can set up a macro command program to be executed from the MANUAL FCTNS Macros screen. After you set up a macro command to run from this screen, you can then select the MANUAL FUNCTIONS menu item and press SHIFT and F3, EXEC to execute the macro command

- The assignment for a MANUAL FUNCTION is MF



## 16.4.3 Input Signal Assignment

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) and, in some cases, a user operator panel input (UI).

**NOTE:** By default, you can assign up to ten macro commands as input signals, or UOP input signals. You can change the number of signals by modifying the value of the \$MACROMAXDRI system variable.

- For digital input signals, indexes 0 through 999 (or the number of digital input signals configured on your system) are available.
- For robot input signals, indexes 0 through the number of robot input signals configured on your system are available. In some cases, this is limited to a maximum of 24.
- If UOP input signals are available, indexes 0 through the number of UOP input signals configured on your system are available.
- For all signals, an index of zero indicates that no Macro is assigned

**NOTE:** Input signals must be properly configured for the Macro command to execute.

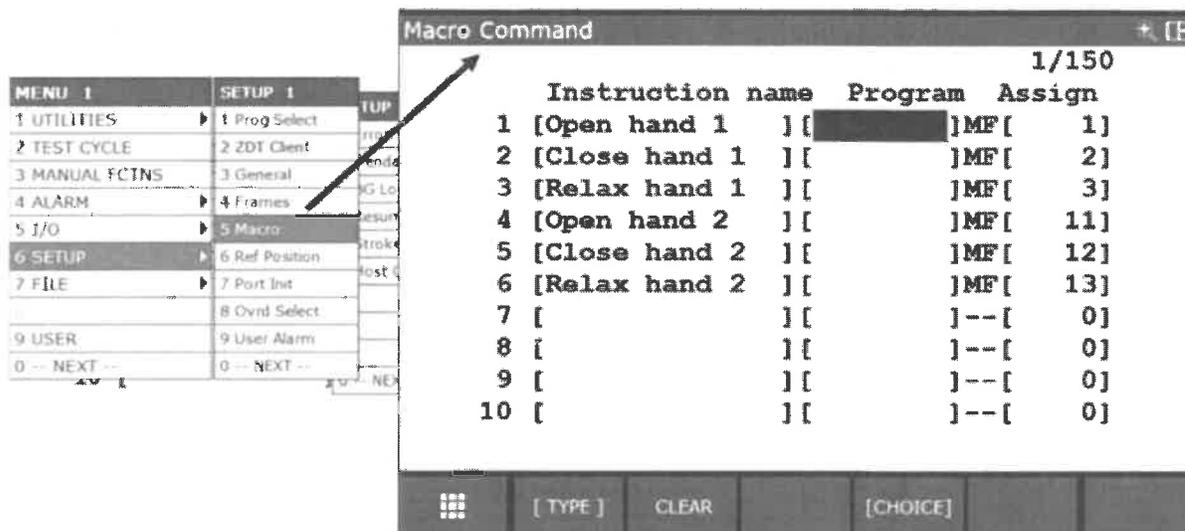
## 16.4.4 Assigning the Macro

As previously stated, a Macro is not a truly functioning Macro until it has been assigned to the macro table. As noted above, there are a number of assignments available. Any one program can be assigned to multiple Macro execution methods. For example, you can map the Home program to run in Manual Mode with a SU [x] assignment, and execute in Automatic Mode with a DI [x] assignment.

 **NOTE:** Prior to assigning a Macro, the program must first be created.

## Procedure 16-2 Making a Macro Assignment

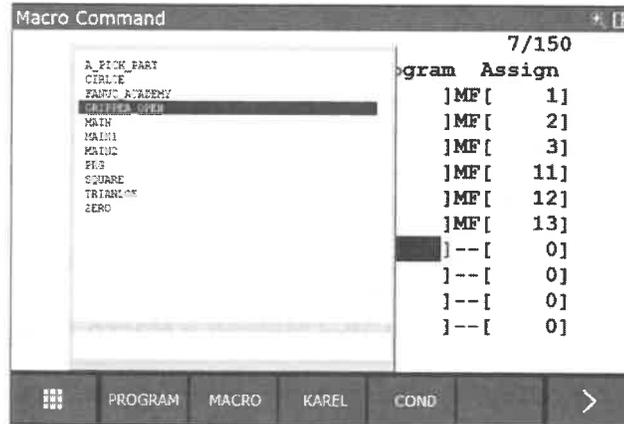
- 1 Press Menu.
  - a Cursor down to SETUP.
  - b On the fly out menu cursor down to MACRO and press ENTER.



- 2 There may be default items listed on the Macro Command Table, if the Program column is blank, these are unassigned.
  - a These can be assigned by associated a program in the program column.

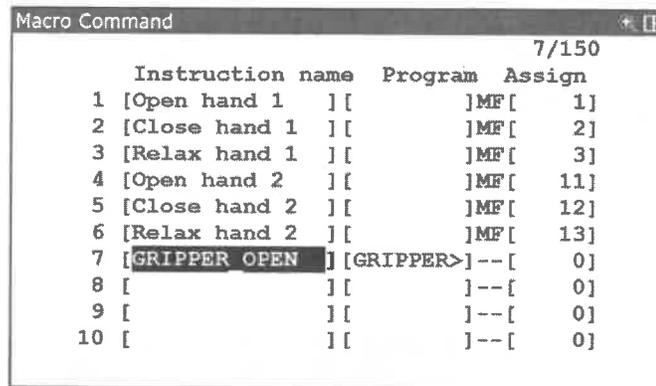
 **NOTE:** If there are already assignments made on the Macro Table and you change them, you may change how your system functions.

- 3 Place your cursor on an open line in the program column.
- 4 With the cursor on the line to make the Macro assignment, press F4, [CHOICE].
  - a Select the program from the list.
  - b If your program is not listed, press F2, MACRO to display Macro designated programs.
  - c With the cursor highlighting the target program, press ENTER.



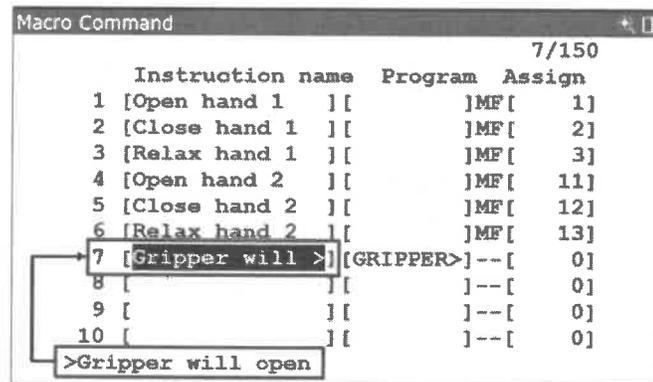
- 5 The name of the selected program will appear in the program column and will also fill the Instruction Name column

**NOTE:** If you place the same program on the Macro Table multiple times and the name already exists in the Instruction Column, this field will be left blank. You will see a prompt on the bottom of the Teach Pendant “the same macro name already exists”.



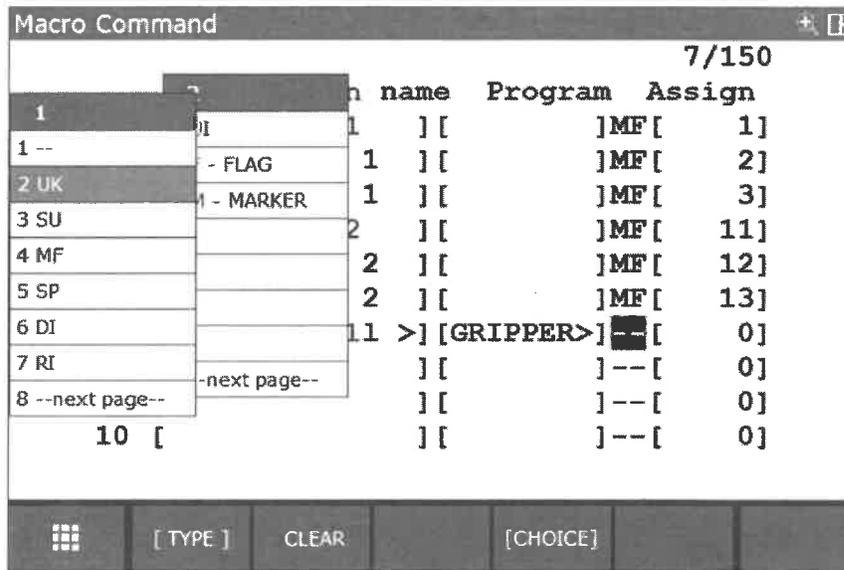
- 6 You should make the instruction name very descriptive as to the function of the program.
  - a There are no naming convention rules when creating an instruction name

**NOTE:** Instruction name is displayed on the MANUAL FCTNS table as well as when using the MACRO Call instruction.



- 7 Cursor to the Assign column.
  - a This is a two part assignment.
- 8 Place your cursor in the first Assign column
  - a Press F4, [CHOICE]
  - b Choose the assignment type.

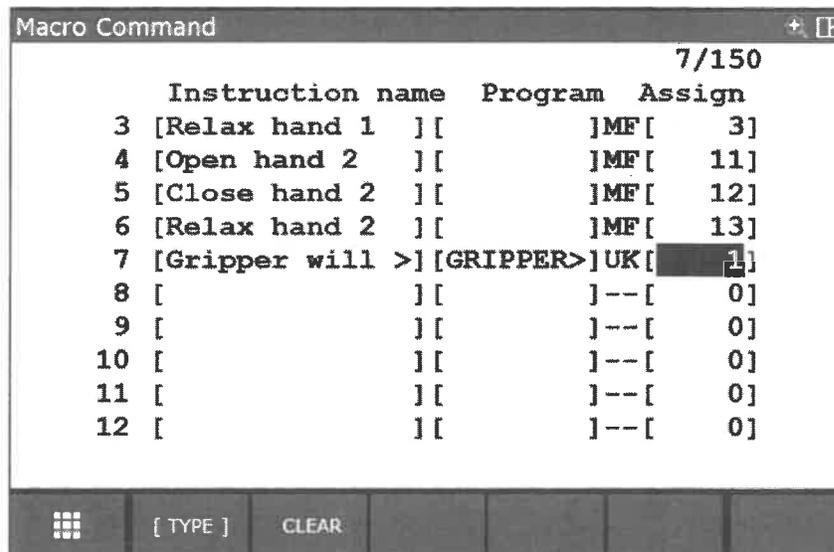
 **NOTE:** If making a UK assignment ensure the Group Mask is [\*,\*,\*,\*,\*,\*]



|                 | name         | Program          | Assign   |
|-----------------|--------------|------------------|----------|
| 1               |              |                  | ]MF[ 1]  |
| 1 --            | - FLAG       | 1 ]              | ]MF[ 2]  |
| 2 UK            | - MARKER     | 1 ]              | ]MF[ 3]  |
| 3 SU            |              | 2 ]              | ]MF[ 11] |
| 4 MF            |              | 2 ]              | ]MF[ 12] |
| 5 SP            |              | 2 ]              | ]MF[ 13] |
| 6 DI            |              | 1.1 >] [GRIPPER> | ]MF[ 0]  |
| 7 RI            |              | ] [              | ]--[ 0]  |
| 8 --next page-- | -next page-- | ] [              | ]--[ 0]  |
| 10              | [            | ] [              | ]--[ 0]  |

Buttons: [TYPE] CLEAR [CHOICE]

- 9 Place your cursor in the second Assign column
  - a Enter the assignment number.



| Instruction name                 | Program | Assign   |
|----------------------------------|---------|----------|
| 3 [Relax hand 1 ]                | ] [     | ]MF[ 3]  |
| 4 [Open hand 2 ]                 | ] [     | ]MF[ 11] |
| 5 [Close hand 2 ]                | ] [     | ]MF[ 12] |
| 6 [Relax hand 2 ]                | ] [     | ]MF[ 13] |
| 7 [Gripper will >] [GRIPPER>] UK | ] [     | ]MF[ 1]  |
| 8 [                              | ] [     | ]--[ 0]  |
| 9 [                              | ] [     | ]--[ 0]  |
| 10 [                             | ] [     | ]--[ 0]  |
| 11 [                             | ] [     | ]--[ 0]  |
| 12 [                             | ] [     | ]--[ 0]  |

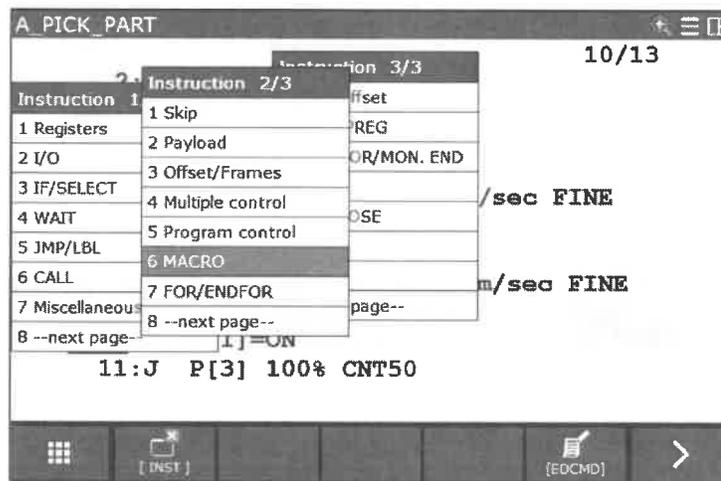
Buttons: [TYPE] CLEAR

## 16.5 Macro Command Instruction

Once a Macro assignment is made on the Macro Table, the Macro can also be called in a program via the 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. This instruction executes the same way as a regular CALL program instruction covered earlier. The only difference is how the line appears in the program.

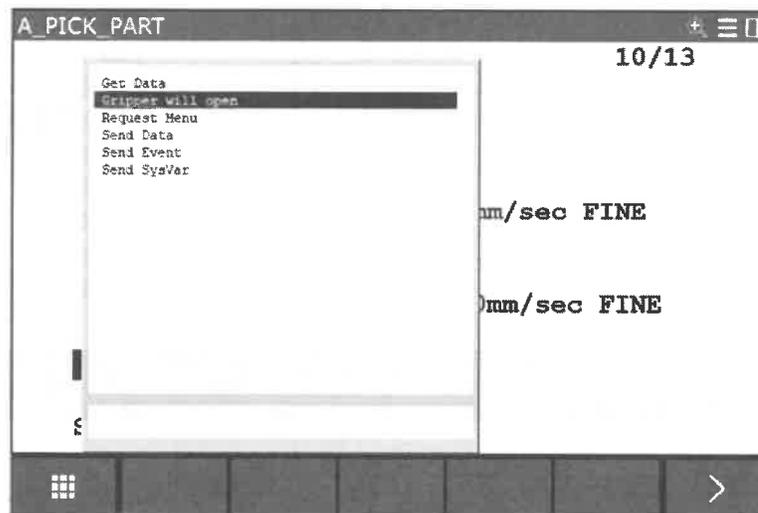
### Procedure 16-3 Programming a Macro Command Instruction

- 1 Place your cursor on the line in a program you wish to add the instruction.
- 2 Press F1, [INST].
- 3 If F1 does not display [INST], press NEXT.
- 4 On the [INST] menu cursor to page 2, and select MACRO.



- 5 Choose the program.

**NOTE:** The programs are listed by Instruction Name as listed on the Macro Table.



- 6 The instruction will be added to the program.
  - a This line executes the exact same as a call.

```
A_PICK_PART 10/13
2: UTOOL_NUM=1
3:
4:J PR[1] 100% FINE
5:J P[1] 100% CNT50
6:L @P[2:Pick Pos] 250mm/sec FINE
7: RO[1]=OFF
8:J P[3] 100% CNT50
9:L @P[4:Place Pos] 250mm/sec FINE
10: Gripper will open
11:J P[3] 100% CNT50
```

## 16.6 Chapter Review

1. **Which Macro Assignments will execute a Macro in Manual Mode?**
  - A DI, RI, UI, SOP
  - B UK, SU, MF
  - C Macros can only run in AUTOMATIC Mode
  - D FANUC software does not support Macros
  
2. **When using a Macro Command Instruction, the program is called by what name?**
  - A Instruction Name
  - B Program Name
  - C Macro Command Instructions do not exist
  - D Comment
  
3. **In order to assign a Macro to a User Key (UK), the group mask must be set to what?**
  - A Any assignment
  - B [1, \*, \*, \*, \*, \*, \*, \*]
  - C [\*, \*, \*, \*, \*, \*, \*, \*]
  - D [1, 1, 1, 1, 1, 1, 1, 1]

**Lab 29**  
**Macro Assignments**

---

**Student Name:** \_\_\_\_\_

**Assignment:** The student will:

- Create a Macro
  - Assign Macros
- 

**Condition:** Given a FANUC Robot and a controller

---

- Step:**
- 1 Create three Macro Programs: (**See Section 16.3 Creating a Macro, pg. 275**)
    - a HOME
    - b REPAIR
    - c SAFE
  - 2 In each respective program record the applicable PR that was created during Lab 18.
  - 3 Assign HOME as:
    - a A MANUAL FUNCTION (**See Section 16.4.2 MANUAL FUNCTIONS Assignments, pg. 278**)
    - b A Digital Input. (**See Section 16.4.3 Input Signal Assignment, pg. 278**)
  - 4 Assign REPAIR as:
    - a A Shift User Key. (**See Section 16.4.1 Teach Pendant User Key Assignments, pg. 277**)
    - b A Digital Input.
  - 5 Assign SAFE as:
    - a A Shift User Key.
    - b A Digital Input.
  - 6 Assign A\_GRIP\_OPEN as UK [1], with an instruction name of 'Gripper will Open'. (**See Section 16.4.4 Assigning the Macro, pg. 279**)
  - 7 Assign A\_GRIP\_CLOSE as UK [2], with an instruction name of 'Gripper will Close'.

8 Test all assignments.

Completed:

Instructor: \_\_\_\_\_

## 17 PR OFFSETS AND MISC. INSTRUCTIONS

### 17.1 Chapter Objectives

- Apply Position Register Instructions
- Apply Position Register [i, j] Instructions
- Apply Position Register Offset Motion Option Instructions
- Utilize Miscellaneous Instructions

### 17.2 Position Registers Overview

Position registers can be used to store global positions, such as a home or a maintenance position which contains X, Y, Z, W, P, and R, positional data.

Position Registers allow positions to be predefined for shared use by many programs. Without additional software options you can define up to 372 Position Registers on an R-30iB PLUS controller.

 **NOTE:** If the controller contains multiple motion groups the number of PR's per group may be less depending on how many motion groups are present.

### 17.3 Position Registers Instructions

Position Register instructions can be used to manipulate the robot position or to store and set position values. These instructions include assignment, addition, and subtraction functions.

As stated earlier, a Position Register stores positional information (X, Y, Z, W, P, R, and configuration). The position information can be a location where the robot moves when used in a motion instruction, or Position Registers can also be used as a storage bank for X, Y, Z, W, P, R values, frame data, joint angles, and offset data. Position Registers are identified by numbers.

There are two kinds 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:

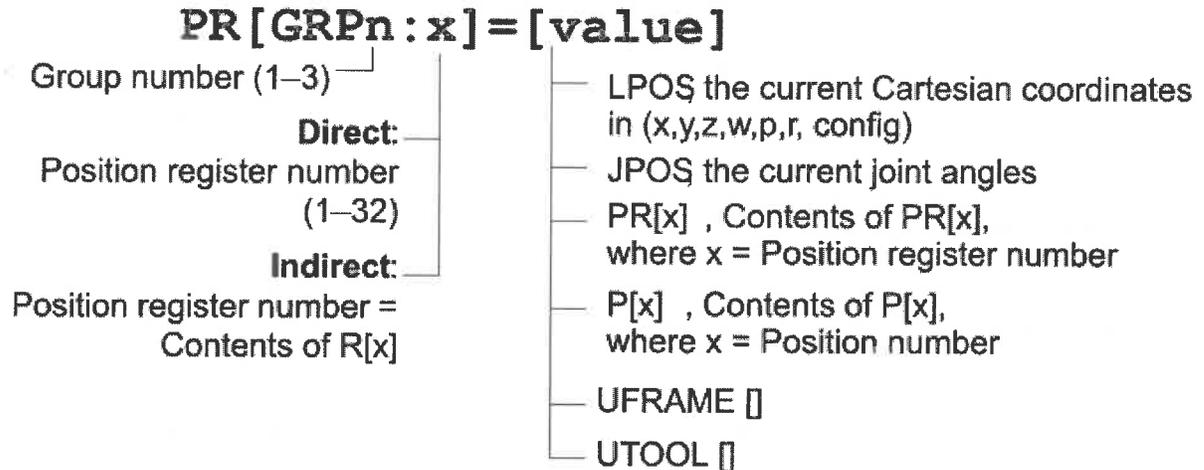
- Whether the group mask will be used. If the group mask is not used, the position register instruction affects the default group only.
- The group or groups that the position register instruction will affect.

# 17: PR Offsets and MISC.

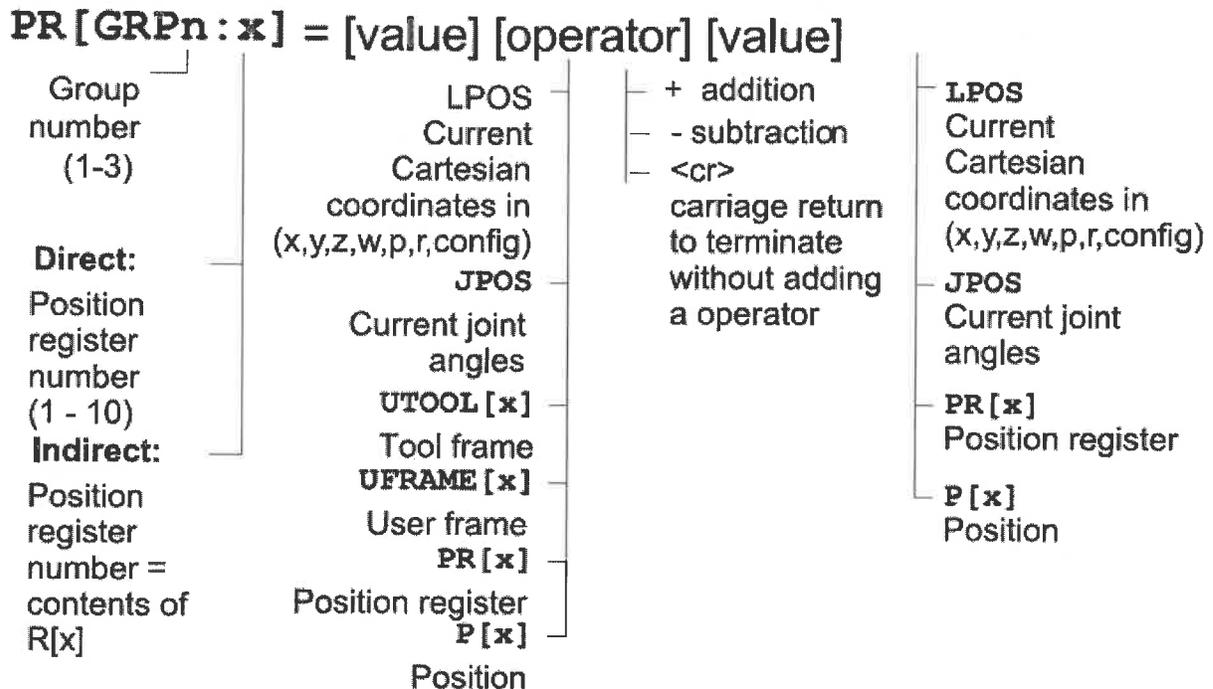
## 17.3.1 PR[x] Position Register Instructions

PR [x] position register instructions manipulate the position register. They include assignment and arithmetic instructions.

An assignment instruction is below. In this instruction the value of the item on the right is set to the PR identified on the left.

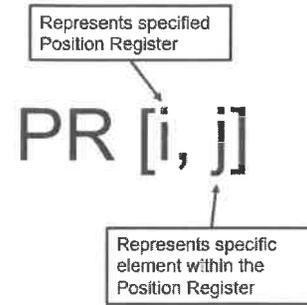


An arithmetic instruction is below. In this type of PR Instruction you are able to add or subtract the contents of two items and place the outcome into a Position Register.

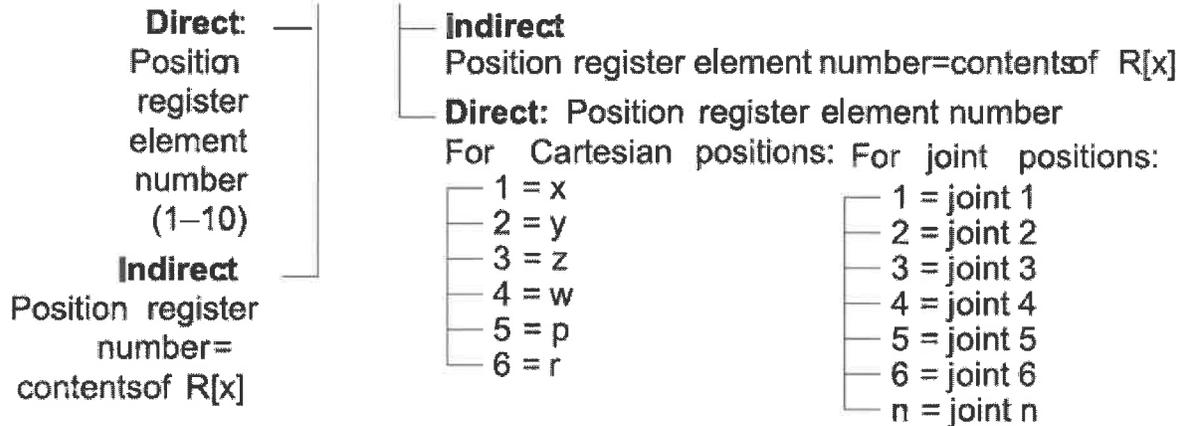


## 17.3.2 PR [i, j] Position Register Element Instructions

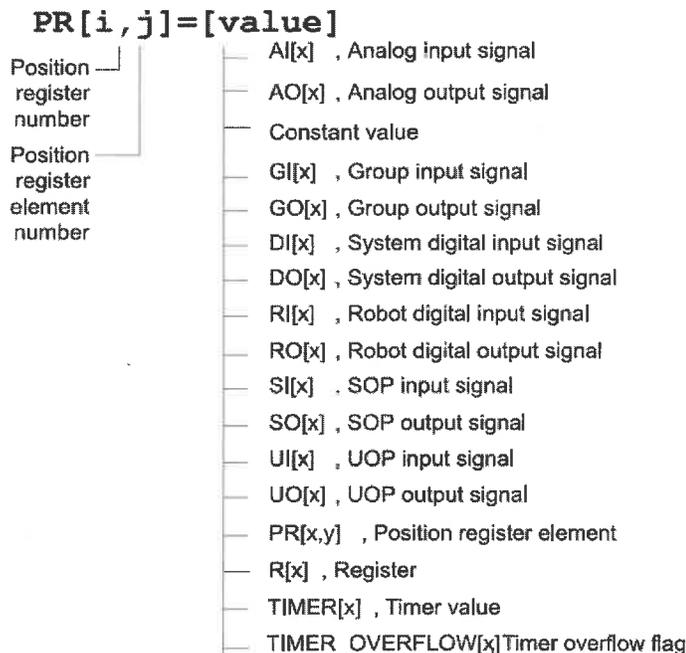
PR [i, j] position register element instructions manipulate a specific position register element. A position register element is one element of a specified position register. In the designation PR [i, j], the "i" represents the position register number and the j represents the position register element. Position register element instructions include assignment and arithmetic instructions.



### PR [i, j]



An assignment instruction is below. In this instruction the value of the item on the right is set to the PR element identified on the left.



# 17: PR Offsets and MISC.

An arithmetic instruction is below. In this type of PR Instruction you are able to perform arithmetic operations and place the outcome into the specified element of a Position Register.

| PR[i,j] [value]                                              | [operator]                           | [value]                                                     |
|--------------------------------------------------------------|--------------------------------------|-------------------------------------------------------------|
| <b>Direct:</b><br>Register number<br>(1 – 32)                | A[x]<br>Analog input                 | + addition                                                  |
|                                                              | AO[x]<br>Analog output               | - subtraction                                               |
| <b>Indirect:</b><br>Register number =<br>contents of<br>R[x] | Constant value                       | * multiplication                                            |
|                                                              | GI[x]<br>Group input                 | / division                                                  |
|                                                              | GO[x]<br>Group output                | DIV whole number division                                   |
|                                                              | DI[x]<br>System digital input        | MOD remainder division                                      |
|                                                              | RI[x]<br>Robot digital input         | <cr> carriage return to terminate without adding a operator |
|                                                              | DO[x]<br>System digital output       |                                                             |
|                                                              | RO[x]<br>Robot digital output        |                                                             |
|                                                              | R[x]<br>Register                     |                                                             |
|                                                              | PR[x,y]<br>Position register element |                                                             |
|                                                              | SI[x]<br>SOP Input                   |                                                             |
|                                                              | SO[x]<br>SOP Output                  |                                                             |
|                                                              | UI[x]<br>UOP Input                   |                                                             |
|                                                              | UO[x]<br>UOP Output                  |                                                             |
|                                                              | TIMER[x]<br>Timer value              |                                                             |
| TIMER_OVERFLOW[x]<br>Timer overflow flag                     |                                      |                                                             |
|                                                              |                                      | A[x]<br>Analog input                                        |
|                                                              |                                      | AO[x]<br>Analog output                                      |
|                                                              |                                      | Constant value                                              |
|                                                              |                                      | GI[x]<br>Group input                                        |
|                                                              |                                      | GO[x]<br>Group output                                       |
|                                                              |                                      | DI[x]<br>System digital input                               |
|                                                              |                                      | RI[x]<br>Robot digital input                                |
|                                                              |                                      | DO[x]<br>System digital output                              |
|                                                              |                                      | RO[x]<br>Robot digital output                               |
|                                                              |                                      | R[x]<br>Register                                            |
|                                                              |                                      | PR[x,y]<br>Position register element                        |
|                                                              |                                      | SI[x]<br>SOP Input                                          |
|                                                              |                                      | SO[x]<br>SOP Output                                         |
|                                                              |                                      | UI[x]<br>UOP Input                                          |
|                                                              |                                      | UO[x]<br>UOP Output                                         |
|                                                              |                                      | TIMER[x]<br>Timer value                                     |
|                                                              |                                      | TIMER_OVERFLOW[x]<br>Timer overflow flag                    |

**17.3.3 Position Register Element Instruction Limitations**

You can use multiple arithmetic operators in a single instruction. However, there are the following limitations:

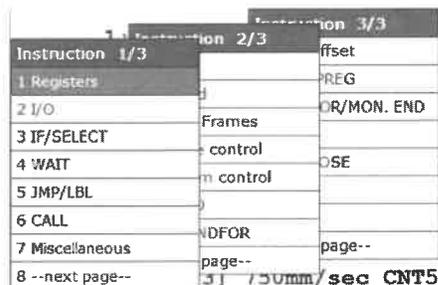
- You can mix + and – in the same instruction. Arithmetic operations within an instruction that mixes + and – will be performed from left to right. You cannot mix \* or / in an instruction that already contains + or –.
- You can mix \* and / in the same instruction. Arithmetic operations within an instruction that mixes \* and / will be performed from left to right. You cannot mix + or – in an instruction that already contains \* or /.
- The maximum number of arithmetic operators you can have in the same instruction is 5.

**17.3.4 Programming a Position Register Instruction**

Whether programming a PR [ ] or PR [i, j] instruction, the steps are very similar. You will be prompted to choose the type of Position Register for the instruction. Once the type is chosen the options that agree with that type of PR will be available.

**Procedure 17-1 Programming a Position Register Instruction**

- 1 Place a blank line in the program at the point to add the Position Register Instruction.
- 2 Press F1, [INST].
  - a If F1 does not display [INST], press NEXT
- 3 Select Register



- 4 Choose the desired arithmetic operation template.
  - a The software will prompt you to fill in the necessary items.
  - b If you choose the incorrect arithmetic operator it can be changed at any time by highlight and pressing F4, [CHOICE].

# 17: PR Offsets and MISC.

| REGISTER statement 1/1 |  |
|------------------------|--|
| 1 ...=...              |  |
| 2 ...=...+...          |  |
| 3 ...=...-...          |  |
| 4 ...=...*...          |  |
| 5 ...=.../...          |  |
| 6 ...=...DIV...        |  |
| 7 ...=...MOD...        |  |
| 8 ...=(...)            |  |

- 5 Choose the required type of Position Register; either PR [ ] or PR [i, j].

| REGISTER statement 1/1 |  |
|------------------------|--|
| 1 R[ ]                 |  |
| 2 PR[ ]                |  |
| 3 PR[i,j]              |  |
| 4 SR[ ]                |  |
| 5                      |  |
| 6                      |  |
| 7                      |  |
| 8 --next page--        |  |

- 6 Input the value of the Position Register to use and if necessary, the element of that register.

**14: PR[... , ...]=...+...**

- 7 Choose the next variable type that the field will utilize.

| REGISTER statement 1/3 |  | REGISTER statement 3/3 |  |
|------------------------|--|------------------------|--|
| 1 R[ ]                 |  | 1 AR[ ]                |  |
| 2 Constant             |  | 2 AO[ ]                |  |
| 3 DO[ ]                |  | 3 AI[ ]                |  |
| 4 DI[ ]                |  | 4 PR[i,j]              |  |
| 5 RO[ ]                |  | 5 SR[ ]                |  |
| 6 RI[ ]                |  | 6                      |  |
| 7 GO[ ]                |  | 7                      |  |
| 8 --next page--        |  | 8 --next page--        |  |

- 8 Quantify the values required by the selected variable.

**14: PR[5, 3]=PR[... , ...]+...**

- 9 Choose the next variable, if necessary.

# 17: PR Offsets and MISC.

| REGISTER statement 1/3 | REGISTER statement 2/3 | REGISTER statement 3/3 |
|------------------------|------------------------|------------------------|
| 1 R[ ]                 |                        |                        |
| 2 Constant             |                        |                        |
| 3 DO[ ]                |                        |                        |
| 4 DI[ ]                |                        |                        |
| 5 RO[ ]                |                        |                        |
| 6 RI[ ]                |                        |                        |
| 7 GO[ ]                | ERFLOW                 |                        |
| 8 --next page--        | R[1]+1                 |                        |

10 Quantify the selected variable.

14: PR[5,3]=PR[5,3]+Constant → 14: PR[5,3]=PR[5,3]+50

**NOTE:** When manipulating values with a Position Register, the data will be relative the active User and Tool Frame.

11 When the program executes, each cycle will add 50mm to the Z element of PR [5].

FANUC\_ACADEMY 8/19

```

7: LBL[1]
8:
9: J P[1:Square] 100% CNT50
10: L P[2] 6000cm/min CNT50
11: L P[3] 750mm/sec CNT50
12: L P[2] 750mm/sec CNT50
13: L P[1:Square] 750mm/sec CNT50
14: PR[5,3]=PR[5,3]+50
15: R[1]=R[1]+1
16: IF R[1]<5, JMP LBL[1]
17:

```

Will add 50mm to the Z element of PR [5]

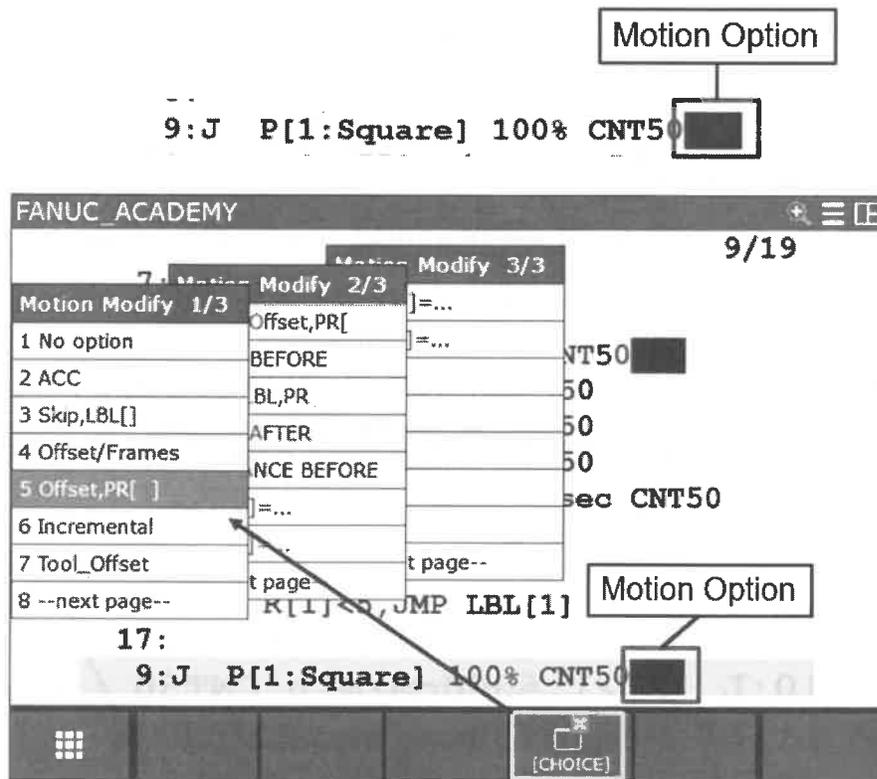
Increments R [1] by one each cycle

Jumps to Label 1 and runs the program again for five cycles then stops.

# 17: PR Offsets and MISC.

## 17.4 OFFSET, PR Motion Option Instruction

The Offset, PR [x] motion option alters positional information by the offset amount specified in the position register PR [x]. When the robot is moving to the stored X, Y, Z, W, P, and R values known as P[1] it also checks the contents of the Offset, PR[#] option for any changes the programmer may want to make to P[1]'s original X, Y, Z, W, P, and R values.



9:J P[1:Square] 100% CNT50  
: Offset,PR[5]

In this case, a Position Register is being used as a storage bank for X, Y, Z, W, P, and R values, joint angles, and offset data. In the program shown above and the positional data shown below P [1] had a Z value of 400mm, with the addition of the 50mm Z value stored in PR [5] the new Z value for P [1] will be 450mm.

| P[1] UF:1 UT:1  |            | CONF:NUT 000 |             |
|-----------------|------------|--------------|-------------|
| X               | 26.000 mm  | W            | 180.000 deg |
| Y               | 180.000 mm | P            | 0.000 deg   |
| Z               | 400.000 mm | R            | 0.000 deg   |
| PR[5] UF:F UT:F |            | CONF:NDB 000 |             |
| X               | 0.000 mm   | W            | 0.000 deg   |
| Y               | 0.000 mm   | P            | 0.000 deg   |
| Z               | 50.000 mm  | R            | 0.000 deg   |

# 17: PR Offsets and MISC.

The values stored in the Position Register can be pre-set in the program or manipulated using program instructions (see previous section). Also, as seen in the previous section specific elements of taught points can be manipulated from within the program using a Position Register Element instruction PR [i, j].

For example the program instruction PR[5, 3] = 100 would change the Z value found in PR [5] to now equal 100mm instead of 50mm in the previous example.

**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.

## 17.4.1 Programming an OFFSET, PR [x] Instruction

When programming OFFSET, PR [x] Instructions it is important to remember the following:

- The offset data in the Position Register is relative to the active USER Frame orientation.
- The offset data is only applied to the motion instruction where it appears.
- If only offsetting in one direction the entire Position Register must be quantified.
  - No elements of the Position Register can be uninitialized.

---

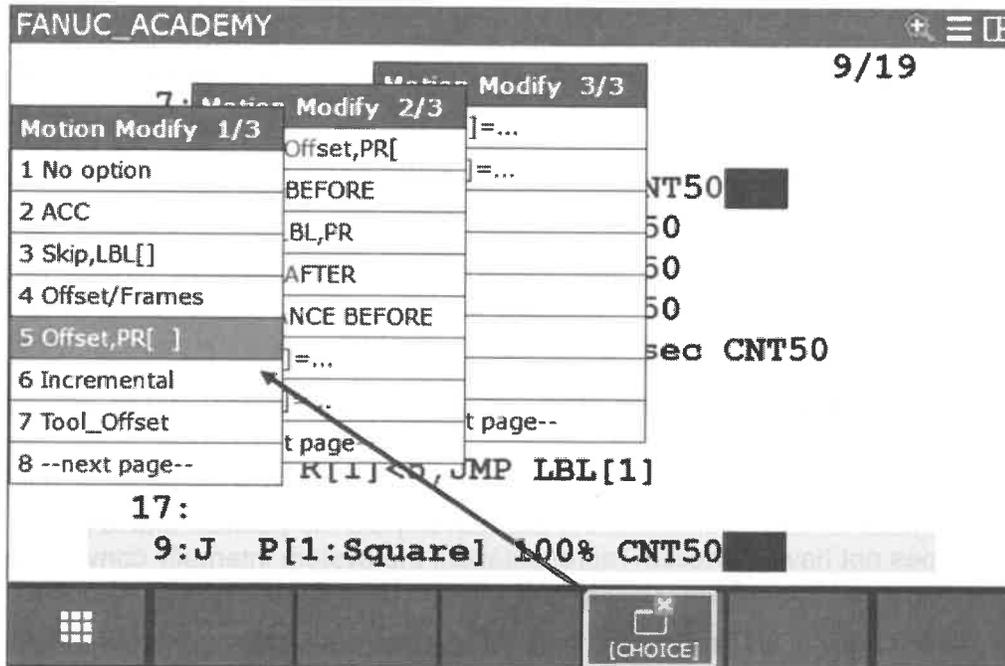
### Procedure 17-2 Programming an OFFSET, PR [x] Motion Option Instruction

---

- 1 Place your cursor on the motion instruction to add the OFFSET, PR [x] motion option.
- 2 Press F4, [CHOICE]
- 3 Choose OFFSET, PR [x] from the menu

 **NOTE:** Choosing TOOL OFFSET, PR [x] will offset the position relative to the Tool Frame Orientation.

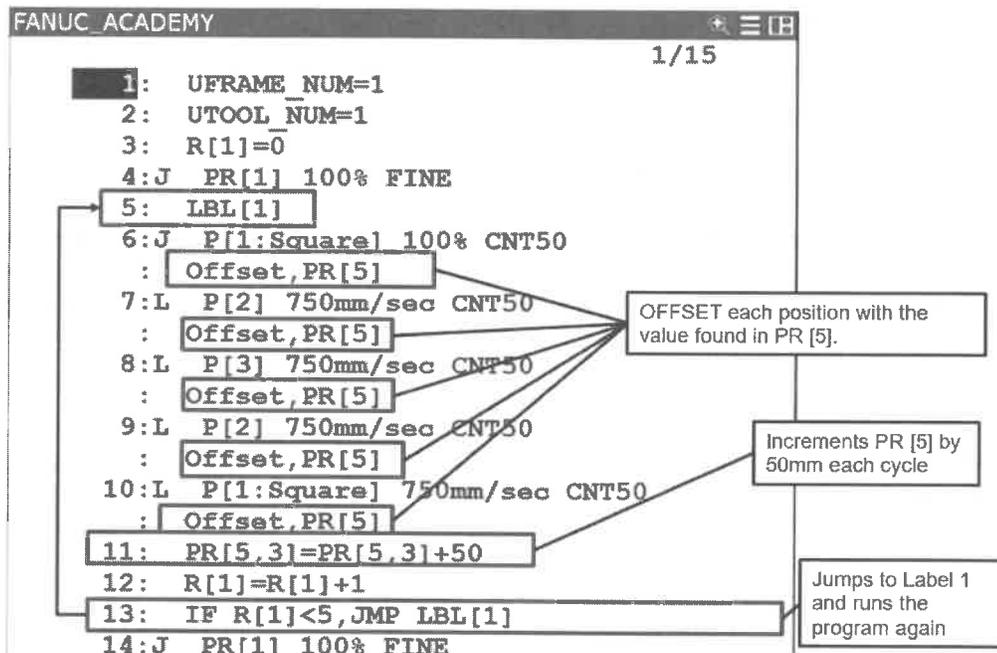
# 17: PR Offsets and MISC.



- 4 Quantify the Position Register that is to be used.

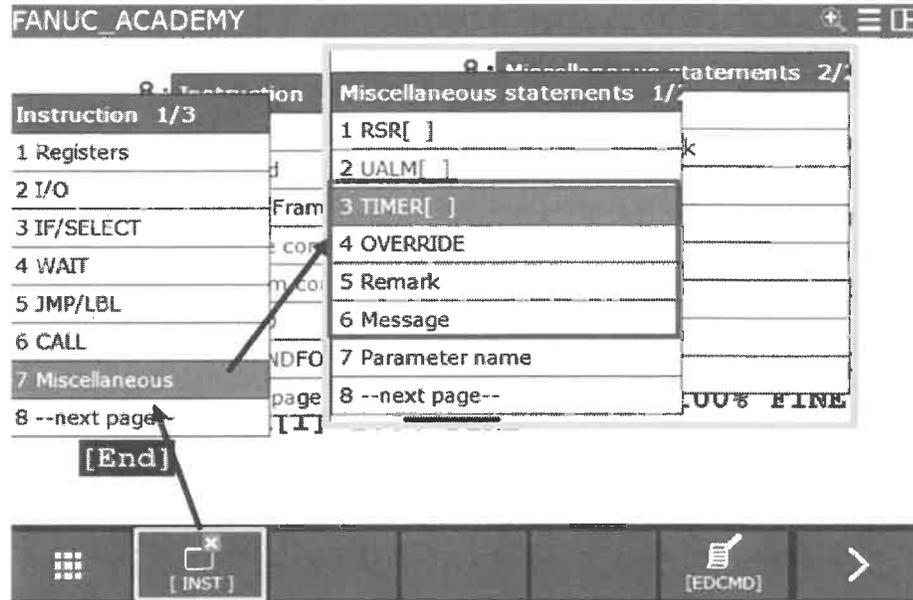
```
9:J P[1:Square] 100% CNT50
: Offset,PR[5]
```

- 5 Repeat for other Motion Instructions, if necessary.
- 6 This program will run five cycles and each cycle will offset in the +Z direction by 50mm.



## 17.5 Miscellaneous Instructions

There are miscellaneous instructions for timer setting, speed override, program remarks, and message handling. All miscellaneous instructions can be programmed by accessing Miscellaneous from the [INST] menu.



### 17.5.1 Timer Instruction

Timer instructions allow you to start, stop, and reset up to 20 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.

**TIMER[x] = [action]**

|                                                |                          |
|------------------------------------------------|--------------------------|
| Direct : _____                                 | START - starts the timer |
| Timer number (1 - 20)                          |                          |
| Indirect : _____                               | STOP - stops the timer   |
| R[x], where timer number =<br>contents of R[x] |                          |
|                                                | RESET - resets the timer |

### 17.5.2 OVERRIDE Instruction

The OVERRIDE instruction sets the speed override to a percentage value of the programmed speed.

**OVERRIDE = x %**

└ Speed override, 0 - 100%

## 17.5.3 Remark Instruction

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 (!).

**!remark text**

The remark instructions can be displayed with a yellow background color to help clarify the program content. The use of color text in the editor can be turned On or Off using the Color command available under the EDCMD function key.

## 17.5.4 Message Instruction

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.

When the MESSAGE [message content] instruction is executed, the user screen is displayed automatically.

**MESSAGE [message content]**

└ Contents of message, from 1 to 23 characters long

**17.6 Chapter Review**

- 1. In the Instruction PR [i, j], what does the “i” represent?**
  - A The Position Register to use
  - B The specific element of the Position Register
  - C The value to increment the Position Register
  - D PR [i, j] is not a valid instruction
  
- 2. The motion option OFFSET, PR [x] is offset relative to the orientation of what Frame?**
  - A TOOL
  - B JOG
  - C WORLD
  - D USER
  
- 3. How many Program Timers are there?**
  - A 5
  - B 10
  - C 15
  - D 20

## Lab 30

### Position Register Offsets

Student Name: \_\_\_\_\_

Assignment: The student will:

- Use OFFSET, PR [x] Instructions
- Use the PR [i, j] instruction

Condition: Given a FANUC Robot and a controller

- Step:**
- 1 Copy A\_COUNTER and rename the program A\_OFFSET. (See **Section 11.3.1 Select Screen Soft Keys, pg. 161**)
  - 2 Modify the program to offset a specific distance above the current path and run a specific number of cycles.
  - 3 Assign a position register to store the positional offset and manually enter a value of zero in all six elements at the DATA screen.
  - 4 Initialize both registers (counter and offset – use same counter register from previous program) outside main loop.
  - 5 Increment counter register within each loop. (See **Section 15.5.2 Performing Arithmetic Operations with Register Instructions, pg. 253**)
  - 6 Increment the Z element of offset register by 15mm on each cycle. (See **Section 17.3.4 Programming a Position Register Instruction, pg. 291**)
  - 7 Use a conditional branching instruction to monitor the number of cycles executed while the program runs. (See **Section 15.6.4 Programming an IF Instruction, pg. 258**)
  - 8 The program syntax will be similar to below.

```
1: UFRAME_NUM=1 ;
2: UTOOL_NUM=1 ;
3: ;
4: PR[5,3]=0 ;
5: R[1:Cycle Count]=0 ;
6: ;
7: J PR[1:HOME] 50% FINE ;
8: ;
9: LBL[1] ;
```

## 17: PR Offsets and MISC.

10: ;  
11:L P[1:Square] 1500mm/sec CNT25 Offset, PR[5] ;  
12:L P[2] 1500mm/sec CNT88 Offset, PR[5] ;  
13:J P[3] 50% FINE Offset, PR[5] ;  
14:L P[4] 1000mm/sec CNT100 Offset, PR[5] ;  
15:L P[1:Square] 1500mm/sec CNT25 Offset, PR[5] ;  
16: ;  
17: PR[5,3]=PR[5,3]+15 ;  
18: R[1:Cycle Count]=R[1:Cycle Count]+1 ;  
19: ;  
20: IF R[1:Cycle Count]<5,JMP LBL[1] ;  
21: ;  
22:J PR[1:HOME] 50% FINE ;

Completed:

Instructor: \_\_\_\_\_

## Lab 31

### Final Exercise

Student Name: \_\_\_\_\_

Assignment: The student will:

- Apply all of the concepts discussed in the course

Condition: Given a FANUC Robot and a controller

- Step: 1 Copy A\_PICK\_PART and rename it A\_MULT\_PICK.
- 2 Create a Position Register to hold offset data that will position the robot directly above the pick position (PR [6]) and/or place position (PR [7]).

 **NOTE:** The offset data is relative to the active USER Frame

```
DATA Position Reg
PR[10] UF:F UT:F CONF:NDB 000
X 0.000 mm W 0.000 deg
Y 0.000 mm P 0.000 deg
Z 50.000 mm R 0.000 deg
Position Detail
PR[5:]=R
PR[6:]=*
PR[7:]=*
PR[8:]=*
PR[9:]=*
PR[10: Above Pick/Place]=*
PR[11:]=*
Enter value
```

- 3 Replace both of the existing “Above Pick” local positons with PR [6: Pick Pos]
- a Add the offset motion option to these lines, referencing the PR you established in **Step 2**.
- 4 Replace both of the existing “Above Place” local positons with PR [7: Place Pos]
- a Add the offset motion option to these lines, referencing the PR you established in **Step 2**.
- 5 Based on the template provided by your instructor, relative to your USER Frame orientation, determine the distance and direction (X, Y, or Z) to offset the first pick positon (PR [6]) to the second part.
- 6 Do the same for the drop position with PR [7].

# 17: PR Offsets and MISC.

- 7 Add the PR [i, j] instruction after each pick/place to offset the next pick/place.
- 8 Replace the existing I/O instructions that either open or close the gripper with your gripper open and / or gripper close programs.
  - a Utilize the MACRO Command Instruction to accomplish this.
- 9 Set payload schedules as needed.
  - a When a part is picked set payload of "tool w/ part".
  - b When a part is placed set payload of "tool only".
- 10 Add a PR [i, j] Instruction that will set the initial value of the pick and place PR that will be adjusted. Example, if you must offset in the X direction set the initial X value in the PR [i, j] instruction.
- 11 This program will run for three cycles.
  - a Add a Data Register to count the three cycles.
  - b Add a conditional instruction that will end the program after three cycles.
- 12 Track the cycle time for the entire program.
  - a Set a Data Register to hold the total cycle time.
- 13 Add remarks where necessary to describe what the program is performing.
- 14 Utilize your existing Group Output to reflect the part number being picked and placed.
  - a For example, if part 1 is being picked set the group output to a value of 1. When part two is being picked set the group output to a value two.
- 15 A template is provided on the following page, if needed.

Completed:

Instructor: \_\_\_\_\_

```
1: UFRAME_NUM=1
2: UTOOL_NUM=1
3:
4: TIMER [1]: RESET
5: TIMER [1]: START
6: PR [6, 1] = (Initial X Value)
7: PR [7, 1] = (Initial X Value)
8: R [5: Cycle Count] = 0
9: PAYLOAD [1]
10:
11: J PR[1:HOME] 100% FINE
12: LBL [1]
13:
14: J PR [6: Pick Pos] 100% CNT50 Offset, PR [10]
15: L PR [6:Pick Pos] 250mm/sec FINE
16: Gripper will Close
17: PAYLOAD [2]
18: R [5: Cycle Count] = R [5: Cycle Count] + 1
19: GO [1] = R [5]
20: J PR [6: Pick Pos] 100% CNT50 Offset, PR [10]
21: PR [6, 1: Pick Pos] = PR [6, 1: Pick Pos] + 65
22:
23: J PR [7: Place Pos] 100% CNT50 Offset, PR [10]
24: L PR [7:Place Pos] 250mm/sec FINE
25: Gripper will Open
26: PAYLOAD [1]
27: J PR [7: Place Pos] 100% CNT50 Offset, PR [10]
28:
29: PR [7, 1: Place Pos] = PR [7, 1: Place Pos] + 65
30: IF R [5: Cycle Count] < 3, JMP LBL [1]
31:
```

32: TIMER [1]: STOP  
33: R[10: Cycle Time]= TIMER [1]  
34:  
35: J PR[1:HOME] 100% FINE



**18 POST TEST**

Name: \_\_\_\_\_ Date: \_\_\_\_\_

This test is designed to evaluate your knowledge of FANUC America Corporation HandlingTool Operations and Programming after taking this course. Each question has four possible answers, choose the one that best answers the question. There is only one right answer. We appreciate you're taking the time to complete it.

Score: \_\_\_ / 100%

- 1. Enabling the iPendant gives motion control solely to whom?**
  - A The PLC
  - B The Controller
  - C The Operator (person with the pendant)
  - D Andy from accounting
- 2. Assuming standard setup, if you jog the robot in World -Y which direction will the TCP move?**
  - A Straight out front
  - B Up
  - C To the robots right
  - D To the robots Left
- 3. What key is depressed to switch between the different jog methods?**
  - A POSN
  - B MENU
  - C FCTN
  - D COORD
- 4. What two buttons display the Jog Menu**
  - A Shift and Display
  - B Menu and any jog key
  - C Shift and COORD
  - D Group and COORD

# 18: Post Test

5. **What type of memory is saved when performing an Image Backup?**
  - A F-ROM Only
  - B SRAM Only
  - C F-ROM and SRAM
  - D DRAM
6. **Singularity error can be corrected by moving?**
  - A Joint 2 (+/-) 10 degrees
  - B Joint 3 (+/-) 15 degrees
  - C Joint 5 (+/-) 10 degrees
  - D Joint 6 (+/-) 15 degrees
7. **Where is the default Tool Center Point?**
  - A Intersection of J1 and J2
  - B Center of the Faceplate
  - C Tip of the default tool
  - D FANUC robots do not use Tool Center Points
8. **What type of I/O handles communication between the controller and the robot?**
  - A Digital I/O
  - B UOP I/O
  - C Robot I/O
  - D SOP I/O
9. **Positional data is dependent on what two frames?**
  - A World and User
  - B Jog and Tool
  - C User and Tool
  - D Tool and World
10. **Which type of instruction defines for the robot how to move, where to move, how fast to move, and how to stop?**
  - A Move Instruction
  - B Logic Instruction
  - C Motion Instruction
  - D Kinematics Instruction

- 11. Which type of pasting will paste the motion instructions in order and renumber the Position ID's based on first available Position ID?**
- A LOGIC
  - B POSITION
  - C POS\_ID
  - D R\_POS
- 12. Which Mode limits all Cartesian speed to less than 250 mm/sec and joint speed to less than 10%?**
- A Auto Mode
  - B T2 Mode
  - C T1 Mode
  - D Neither, this is a system variable setting
- 13. Which motion type moves the TCP in a straight line from the start position to the destination position?**
- A Linear
  - B Joint
  - C Circular
  - D Spline
- 14. Which Position must the DEADMAN Switch be held to enable motion?**
- A Released
  - B Centered
  - C Fully Depressed
  - D The iPendant does not have a DEADMAN Switch
- 15. Before forcing an input it must be \_\_\_\_\_?**
- A Simulated
  - B Named
  - C Part of a Group
  - D Inputs can never be forced

## 18: Post Test

16. What type of memory is saved when performing a File Backup?
- A F-ROM Only
  - B SRAM Only
  - C F-ROM and SRAM
  - D DRAM
17. The six point method of defining a Tool Frame defines the Location and \_\_\_\_\_ of a Tool Frame.
- A Distance
  - B Orientation
  - C Operation
  - D Offset
18. What are the two possible actions of an IF/SELECT statement?
- A Select & Set an output
  - B Set an Output & CALL
  - C JMP LBL & CALL
  - D JMP LBL & WAIT
19. The motion option OFFSET, PR [x] is offset relative to the orientation of what Frame?
- A TOOL
  - B JOG
  - C WORLD
  - D USER
20. Which Macro Assignments will execute a Macro in Manual Mode?
- A DI, RI, UI, SOP
  - B UK, SU, MF
  - C Macros can only run in AUTOMATIC Mode
  - D FANUC software does not support Macros



The products in this publication may be controlled based on Japan's "Foreign Exchange and Foreign Trade Law". The export from Japan may be subject to an export license from the government of Japan. Additionally, the products also may be controlled by the export or re-export regulations of the United States government or other countries.

If any customer intends to, or shall, export (or re-export), directly or indirectly, any portion of the products, technology or software relating thereto, it is the responsibility of customer to assure compliance with United States and other export control laws and regulations and if appropriate, to secure any required export licenses or approvals in its own name.



Follow us on Social Media:



FANUC America Corporation  
[www.fanucamerica.com/](http://www.fanucamerica.com/)  
888-FANUC-US  
MATAGHAND0318CE

© 2018 FANUC America Corporation. All rights reserved